

ANALISIS QOS *VIDEO STREAMING* PADA JARINGAN

***WIRELESS* MENGGUNAKAN METODE HTB**

(HIERARCHICAL TOKEN BUCKET)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika



oleh :

DONI HELTON JANIUS

1 0 6 5 1 0 0 4 3 7 0

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU**

2013

QOS VIDEO STREAMING ANALYZING OVER WIRELESS NETWORK USING HTB METHOD (*HIERARCHICAL TOKEN BUCKET*)

DONI HELTON JANIUS
1 0 6 5 1 0 0 4 3 7 0

Date of Final Exam : Juli 04th 2013

Graduation Period : November 2013

Information Engineering Department

Faculty of Science and Technology

State Islamic University of Sultan Syarif Qasim Riau

ABSTRACT

Coincide with the increase of Internet users in Indonesia, the number of users increases either streaming video or audio. This increase is based on the very rapid development of technology especially notebooks and tablets that use the wireless internet access. The use of streaming video transmission over wireless networks different from wired networks because of the characteristics of wireless network is limited compared with the wired network, and transmission characteristics of video streaming that require different handling than the transmission of text data in general. To maintain the stability internet service to users who access the various applications on the internet, would be applied QOS (Quality Of Service) methods in wireless network with HTB (Hierarchical Token Bucket) data used is a video with the quality of 240p, 360p, 480p, 720p. QoS parameters are analyzed consisted of Delay, Jitter, Packet Loss, and Throughput. Results of analysis using RTP protocol for video streaming jitter value of 0 ms because RTP is designed to compensate for jitter and desequencing, if a packet is dropped or lost in the network then RTP will not perform retransmission in order to avoid delay due to retransmission requests. To get good results in the video, packet loss should range between 0% - 3%. If packet loss is more than 3% video results obtained are not satisfy because the video had a lot of damage to the image and sound.

Keywords : *HTB ,QOS, Video Streaming, Wireless.*

KATA PENGANTAR



Assalamu'alaikum Wr. Wb.

Alhamdulillah, segala puji bagi-Mu ya Allah yang memberikan kemudahan bagi hamba-Nya dalam penyelesaian tugas akhir ini. Shalawat beriring salam kita haturkan kepada Nabi Muhammad SAW beserta keluarga dan sahabat-sahabatnya, yang telah memberikan suri tauladan kepada kita agar dapat menjalankan kehidupan ini dengan memegang prinsip Syariat-Nya.

Penulisan tugas akhir ini merupakan salah satu syarat untuk memperoleh gelar sarjana dan sebagai sarana pengembangan ilmu pada jurusan yang diambil, khususnya pada bidang analisa Jaringan. Tugas akhir ini membahas tentang Analisis *Quality Of Service Video Streaming* pada Jaringan *Wireless* menggunakan Metode HTB (*Hierarchical Token Bucket*).

Ucapan terima kasih disampaikan kepada setiap orang yang terlibat dalam proses penyelesaian tugas akhir ini, semoga Allah memberikan balasan yang setimpal dengan usaha ikhlas yang telah diberikan. Ucapan terima kasih ini, penulis sampaikan kepada:

1. Bapak Prof. Dr. H.M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Ibu Dra. Yenita Morena. M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.

3. Bu DR. Okfalisa, ST, M.Sc selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Sultan Syarif Kasim Riau.
4. Bapak Iwan Iskandar, ST, MT selaku pembimbing yang telah memberikan masukan, kritik dan saran mengenai tugas akhir penulis. Terima kasih banyak atas segala waktu dan perhatian yang bapak berikan.
5. Bapak M.Safrizal, ST,M.Cs, selaku penguji 1 yang telah memberikan kritik dan saran yang membangun pada tugas akhir ini sehingga menjadi suatu karya ilmiah yang bagus.
6. Bapak Reski Mai Candra, ST, M.Sc, selaku penguji 2 yang telah memberikan kritik serta saran yang membangun pada tugas akhir ini sehingga menjadi suatu karya ilmiah yang bagus.
7. Bapak Reski Mai Candra, ST, M.Sc Selaku Koordinator Tugas Akhir, yang banyak membantu proses penerbitan tulisan ini.
8. Seluruh dosen Jurusan Teknik Informatika UIN Suska Riau yang telah memberikan ilmu dan pengetahuan yang bermanfaat kepada penulis selama mengikuti perkuliahan di Jurusan Teknik Informatika.
9. Orang tua penulis, Mama tercinta terima kasih atas doa, dukungan, serta motivasi tiada henti yang diberikan kepada penulis selama masa perkuliahan. Semoga mereka berdua selalu dalam lindungan Allah SWT.
10. Adik-Kakak penulis, Alissa Ifni Tiffany, Hendri Gunawan dan Cloudya Angel Siska yang selalu mendo'akan penulis dan tetap memberikan dukungan kepada penulis untuk menjadi orang yang sukses.

11. Bos penulis, Bapak Edward Manurung dan Bapak Rynhart , yang sangat banyak membantu dalam penyelesaian tugas akhir ini.
12. Sahabat penulis, Mizan Ridho, yang banyak membantu dari awal penelitian hingga tugas akhir ini selesai.
13. Kekasih penulis, Yuniati, yang banyak memberikan dukungan dan motivasi kepada peneliti.
14. Genk Basecamp, Angga Novanda Putra, Selamat Raharjo, Khairul Fitrah, Jomy, Amar, Roni Erwanto, dkk yang banyak membantu dalam menyelesaikan tugas akhir ini.
15. Teman-teman seperjuangan di Teknik Informatika semua angkatan yang telah membantu kelancaran dan dukungan dalam tugas akhir ini.

Tugas akhir ini disadari masih jauh dari kesempurnaan. Penulis berharap mendapatkan masukan dari pembaca atas isi tugas akhir ini.

Akhirnya penulis berharap semoga Tugas Akhir ini dapat memberikan sesuatu yang bermanfaat bagi siapa saja yang membacanya. Amin.

Wassalamu'alaikum Wr. Wb.

Pekanbaru, Juli 2013

Doni Helton Janius

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xii
DAFTAR GAMBAR	xvi
DAFTAR TABEL.....	xviii
DAFTAR LAMPIRAN	xx
BAB I PENDAHULUAN	I-1
I.1. Latar Belakang	I-1
I.2. Rumusan Masalah	I-4
I.3. Batasan Masalah.....	I-4
I.4. Tujuan Penelitian	I-4
I.5. Sistematika Penulisan	I-5
BAB II LANDASAN TEORI.....	II-1
II.1. QOS (<i>Quality Of Service</i>).....	II-1
II.1.1. Pengertian QOS.....	II-1
II.1.2. Parameter QOS.....	II-2
II.1.3. Model Layanan QOS.....	II-5
II.2. TCP/IP	II-9
II.2.1. <i>Internet Protocol (IP)</i>	II-12

II.2.2. <i>User Datagram Protocol (UDP)</i>	II-15
II.2.3. <i>Real-Time Transport Protocol (RTP)</i>	II-16
II.3. <i>Video Streaming</i>	II-17
II.3.1. <i>Konsep Video Streaming</i>	II-17
II.3.2. <i>Protokol Video Streaming</i>	II-18
II.3.3. <i>Multicast Transport pada Aplikasi Video</i>	II-19
II.3.4. <i>Kompresi Audio</i>	II-20
II.3.5. <i>Kompresi Video</i>	II-23
II.4. <i>Wireless</i>	II-24
II.4.1. <i>IEEE 802.11 Standar Wireless LAN</i>	II-25
II.4.1.1. <i>Standar IEEE</i>	II-25
II.4.1.2. <i>Perkembangan dari Standar 802.11</i>	II-25
II.4.1.3. <i>Kelebihan Standar 802.11</i>	II-26
II.4.1.4. <i>Kelemahan Standar 802.11</i>	II-26
II.4.2. <i>Standar IEEE 802.11</i>	II-27
II.4.2.1. <i>IEEE 802.11 a</i>	II-27
II.4.2.2. <i>IEEE 802.11 b</i>	II-28
II.4.2.3. <i>IEEE 802.11 e</i>	II-28
II.4.2.4. <i>IEEE 802.11 g</i>	II-31
II.4.2.5. <i>IEEE 802.11 n</i>	II-31
II.5. <i>HTB (Hierarchical Token Bucket)</i>	II-32
II.5.1. <i>General Schedule HTB</i>	II-33
II.5.2. <i>Estimator</i>	II-33
II.5.3. <i>Metode Pengendalian Traffic</i>	II-35
II.5.4. <i>Action Research</i>	II-39
II.5.5. <i>Tujuan dan Ciri-ciri Pengendalian Tindakan</i>	II-39
BAB III <i>METODOLOGI PENELITIAN</i>	III-1
III.1. <i>Diagnosis (Melakukan Diagnosa)</i>	III-1
III.2. <i>Action Planning (Rencana Tindakan)</i>	III-2

III.3. <i>Action Taking</i> (Melakukan Tindakan)	III-2
III.4. <i>Evaluating</i> (Melakukan Evaluasi)	III-3
III.5. <i>Learning</i> (Pembelajaran)	III-3
BAB IV ANALISA DAN PERANCANGAN	IV-1
IV.1. Analisa Kebutuhan Sistem	IV-1
IV.1.1. <i>Diagnosis</i> (Melakukan Diagnosa)	IV-1
IV.1.2. <i>Action Planning</i> (Rencana Tindakan)	IV-3
IV.1.3. Persiapan Perangkat	IV-3
IV.2. PERANCANGAN	IV-6
IV.2.1. Perancangan Infrastruktur	IV-6
IV.2.2. Topologi Jaringan	IV-7
IV.2.3. <i>IP Address</i>	IV-7
IV.2.4. Skenario Pengujian	IV-8
IV.2.4.1. Skenario I	IV-24
IV.2.4.2. Skenario II	IV-24
IV.2.4.3. Skenario III	IV-24
IV.2.4.4. Skenario IV	IV-25
IV.2.4.5. Skenario V	IV-25
IV.2.4.6. Skenario VI	IV-25
IV.2.4.7. Skenario VII	IV-25
IV.2.4.8. Skenario VIII	IV-26
BAB V IMPLEMENTASI DAN PENGUJIAN	V-1
V.1. Implementasi	V-1
V.1.1. Konfigurasi Jaringan <i>Wireless</i>	V-1
V.1.2. Konfigurasi Pico Station M2HP	V-1
V.1.3. Konfigurasi <i>Server Streaming</i>	V-5
V.1.4. Konfigurasi <i>Client</i>	V-14
V.1.5. <i>Action Taking</i>	V-20
V.1.5.1.1. Langkah Pengambilan Data	V-21

V.2. Pengujian	V-27
V.3. <i>Evaluating</i> (Melakukan Evaluasi)	V-37
BAB VI PENUTUP	VI-1
VI.1. Kesimpulan	VI-1
VI.2. Saran.....	VI-2
DAFTAR PUSTAKA	
DAFTAR RIWAYAT HIDUP	

BAB II

LANDASAN TEORI

II.1. QOS (*Quality Of Service*)

II.1.1. Pengertian QOS

Quality of Service (QoS) adalah kemampuan suatu jaringan untuk menyediakan layanan yang baik dengan menyediakan *bandwidth*, mengatasi *jitter* dan *Delay*. Parameter QoS adalah *latency*, *jitter*, *packet loss*, *throughput*, *MOS*, *echo cancellation* dan PDD. QoS sangat ditentukan oleh kualitas jaringan yang digunakan. Terdapat beberapa factor yang dapat menurunkan nilai QoS, seperti : Redaman, *Distorsi*, dan *Noise* (Jonathan,2011).

Performa jaringan komputer dapat bervariasi akibat dari beberapa masalah, seperti halnya masalah *bandwidth*, *latency* dan *Jitter*, yang dapat membuat efek yang cukup besar bagi banyak aplikasi. Sebagai contoh, *video streaming* dapat membuat pengguna kesal ketika paket data aplikasi tersebut berjalan dengan *bandwidth* yang tidak cukup, dengan *latency* yang tidak dapat diprediksi, atau *Jitter* yang berlebih. Beberapa fitur *Quality of Service* (QoS) dapat menangani masalah diatas, dapat menurunkan *latency* dengan mengendalikan pengiriman paket data dan membatasi paket data tertentu dan *Jitter* dapat diprediksi dan dicocokkan dengan kebutuhan aplikasi yang digunakan di dalam jaringan tersebut.

Teknologi QOS ini adalah teknologi yang memungkinkan administrator jaringan untuk dapat menangani berbagai efek akibat terjadinya kemacetan pada lalu lintas aliran paket dari berbagai layanan yaitu dengan mengatur dan memberikan prioritas pada jaringan tersebut, ini akan mengoptimalkan aplikasi yang kritis atau yang memiliki *Delay* sensitif untuk dapat berjalan sebagaimana mestinya. Dengan implementai QOS, *network administrator* akan memiliki fleksibilitas yang tinggi

untuk ,mengontrol aliran dan kejadian-kejadian yang ada di aliran paket pada jaringan.

II.1.2. Parameter QOS

Performansi mengacu ke tingkat kecepatan dan keandalan penyampaian berbagai jenis beban data di dalam suatu komunikasi. Performansi merupakan kumpulan dari beberapa parameter teknis (Jonathan,2011) yaitu :

1. *Delay*

Delay Adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama (Jonathan,2011).

Delay versi *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) dikelompokkan menjadi empat kategori seperti terlihat pada Tabel 2.1.

$$Delay \text{ rata-rata} = \frac{\text{Total Delay}}{\text{Total Paket yang diterima}} \dots\dots\dots (II.1)$$

Tabel 2.1. Kategori *Delay*

Kategori Degradasi	<i>Delay</i>
Sangat Bagus	<150ms
Bagus	150 ms s/d 300 ms
Sedang	300 ms s/d 450 ms
Buruk	>450 ms

2. *Jitter*

Hal ini diakibatkan oleh variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu penghimpunan ulang paket-paket di akhir perjalanan *jitter*. *Jitter* lazimnya disebut variasi *Delay* ,berhubungan erat dengan *latency*, yang menunjukkan banyaknya variasi *Delay* pada taransmisi data di

jaringan. *Delay* antrian pada router dan switch dapat menyebabkan *jitter* (Jonathan,2011).

Kategori kinerja jaringan berbasis IP dalam *Jitter* versi *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) mengelompokkan menjadi empat kategori penurunan kinerja jaringan berdasarkan nilai *Jitter* seperti terlihat pada Tabel 2.2.

$$\text{Jitter} = \frac{(\text{Total variasi delay})}{\text{Total paket yang diterima} - 1} \dots\dots\dots (\text{II.2})$$

Tabel 2.2. Kategori *Jitter*

Kategori Degradasi	Peak <i>Jitter</i>
Sangat Bagus	0 ms
Bagus	0 ms s/d 75 ms
Sedang	75 ms s/d 125 ms
Buruk	125 ms s/d 225 ms

3. *Packet Loss*

Merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan dan hal ini berpengaruh pada semua aplikasi karena *re-transmisi* akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi-aplikasi tersebut. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi kongesti yang cukup lama, *buffer* akan penuh, dan data baru tidak akan diterima (Jonathan,2011).

Packet loss Versi *Telecommunications and Internet Protocol Harmonization Over Networks* (TIPHON) dikelompokkan menjadi empat kategori seperti terlihat pada Tabel 2.3.

$$\text{Packet Loss} = \frac{(\text{paket dikirim} - \text{paket diterima})}{\text{Paket dikirim}} \times 100\% \dots\dots\dots (\text{II.3})$$

Tabel 2.3. Kategori Packet Loss

Kategori Degradasi	<i>Packet Loss</i>
Sangat Bagus	0 %
Bagus	3%
Sedang	15%
Buruk	25%

4. *Throughput*

Throughput adalah kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Troughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada *destination* selama *interval* waktu tertentu dibagi oleh durasi *interval* waktu tersebut (Jonathan,2011).

$$\text{Throughput} = \frac{\text{jumlah data yang diterima}}{\text{waktu pengiriman data}} \dots\dots\dots (\text{II.4})$$

Quality of Service digunakan untuk mengukur tingkat kualitas koneksi jaringan TCP/IP internet atau intranet. Penyebab Qos yang buruk yaitu Redaman, *Distorsi* dan *Noise*. Dalam usaha menjaga dan meningkatkan nilai QoS, dibutuhkan teknik untuk menyediakan utilitas jaringan, yaitu dengan mengklasifikasikan dan memprioritaskan setiap informasi sesuai dengan karakteristiknya masing-masing. Contohnya, terdapat paket data yang bersifat sensitif terhadap *Delay* tetapi tidak sensitif terhadap packet loss seperti VoIP, ada juga paket yang bersifat sensitif terhadap packet loss tetapi tidak sensitif terhadap *Delay* seperti transfer data.

II.1.3. Model Layanan Qos

Terdapat 3 tingkat metode QoS yang umum dipakai, yaitu *best-effort service*, *integrated service* dan *differentiated service*.

a. *Best-effort service*

Best-Effort merupakan model servis yang dihantarkan kepada penggunaanya akan dilakukan sebisa mungkin dan sebaik-baiknya tanpa ada jaminan apa- apa. Jika ada sebuah data yang ingin dikirim, maka data tersebut akan di kirim segera begitu media perantaranya siap dan tersedia. Data yang dikirim juga tidak dibatasi, tidak diklasifikasikan, tidak perlu mendapatkan ijin dari perangkat manapun, tidak diberi *policy*, semuanya hanya berdasarkan siapa yang datang terlebih dahulu ke perangkat *gateway*.

Model *Best-Effort* ini tidak memberikan jaminan apa-apa terhadap reliabilitas, performa, *bandwidth*, kelancaran data dalam jaringan, *Delay*, dan banyak lagi parameter komunikasi data yang tidak dijamin. Data akan dihantarkan sebisa mungkin untuk sampai ke tujuannya. Jika hilang ditengah jalan atau tertunda dengan waktu yang cukup lama di dalam perjalanannya, maka tidak ada pihak maupun perangkat yang bertanggung jawab.

Model *Best-Effort* ini sangat cocok digunakan dalam jaringan dengan koneksi lokal (LAN) atau jaringan dengan koneksi WAN yang berkecepatan sangat tinggi. Model ini sangat tepat jika digunakan dalam jaringan yang melewati aplikasi dan data yang bermacam-macam dengan tingkat prioritas yang sama. Jadi semua aplikasi didalamnya memiliki kualitas yang sama.

Jenis QoS ini tidak cocok digunakan untuk melayani aplikasi-aplikasi bisnis yang kritis dan penting, karea aplikasi tersebut biasanya membutuhkan perlakuan istimewa untuk dapat berjalan dengan baik. Untuk membuat QoS model *Best-Effort* ini, biasanya antarmuka *router* atau perangkat jaringan berkemampuan QoS dikonfigurasi dengan menggunakan metode *queing First In First Out* (FIFO). Dengan begitu data yang masuk pertama kali akan keluar pertama kali juga, maka terciptalah servis model *Best-Effort* yang sangat adil dalam hal perlakuannya di perangkat QoS (Jonathan,2011).

b. *Integrated service*

Integrated Service Model atau disingkat *IntServ* merupakan sebuah model QoS yang bekerja untuk memenuhi berbagai macam kebutuhan QoS berbagai perangkat dan berbagai aplikasi dalam sebuah jaringan. Dalam model *IntServ* ini, para pengguna atau aplikasi dalam sebuah jaringan akan melakukan request terlebih dahulu mengenai servis dan QoS jenis apa yang mereka dapatkan, sebelum mereka mengirimkan data. Request tersebut biasanya dilakukan dengan menggunakan sinyal-sinyal yang jelas dalam proses komunikasinya.

Dalam request tersebut, pengguna jaringan atau sebuah aplikasi akan mengirimkan informasi mengenai profile *traffic* mereka ke perangkat QoS. Profile traffic tersebut akan menentukan hak-hak apa yang akan mereka dapatkan seperti misalnya berapa *bandwidth* dan *Delay* yang akan mereka terima dan gunakan.

Setelah mendapatkan konfirmasi dari perangkat QoS dalam jaringannya, maka pengguna dan aplikasi tersebut baru diijinkan untuk melakukan transaksi pengiriman dan penerimaan data. Transaksi data akan dilakukan dalam batasan-batasan yang telah diberikan oleh perangkat QoS tersebut tanpa kecuali.

Sebuah perangkat QoS biasanya akan bertindak sebagai pengontrol hak-hak yang akan diterima oleh pengguna. Sedangkan pengguna jaringan dan aplikasi didalamnya bertugas untuk mengirimkan *profile* nya untuk dapat diproses dalam perangkat QoS. Setelah hak-hak pengguna jaringan jelas, perangkat QoS akan memenuhi komitmen yang telah dijanjikannya dengan cara mempertahankan status semua pengguna dan kemudian melakukan proses-proses QoS untuk memenuhinya. Proses-proses tersebut adalah *Packet Classification*, *Policing*, *Queing*, dan banyak lagi yang akan dibahas selanjutnya.

Pada kebanyakan perangkat jaringan yang mampu menjalankan QoS model *IntServ* ini, dilengkapi sebuah sistem sinyaling yang bertugas untuk mengirimkan *profile* dan *request* mereka ke perangkat QoS. Sistem sinyaling tersebut sering disebut dengan istilah *Resource Reservation Protocol* (RSVP).

RSVP merupakan protokol *signaling* khusus untuk keperluan QoS. Protokol ini menggunakan info dari *routing protocol* untuk menentukan jalur terbaik menuju ke suatu lokasi. Meskipun RSVP sangat cocok digunakan untuk keperluan pengaturan QoS pada aplikasi real-time seperti IP *Telephony*, NetMeeting, IPTV *streaming*, dan banyak lagi, namun penggunaan RSVP sangatlah terbatas.

Penggunaan RSVP sangat terbatas dikarenakan semua perangkat yang berada dalam jaringan yang mendukung QoS jenis ini harus mendukung system sinyaling RSVP. Selain itu sistem sinyaling ini juga sangat haus akan proses CPU dan kapasitas memori. Dengan demikian penggunaannya tidak terlalu meluas.

Dengan adanya sebuah router berkemampuan QoS dan disatukan dengan perangkat jaringan yang mendukung RSVP, maka biasanya para penjual jasa jaringan dan internet dapat menciptakan dua jenis servis untuk dijual:

1. Guaranteed Rate Service

Jika diterjemahkan arti dari servis ini adalah data rate yang digaransi. Maksud dari servis ini adalah pihak penyedia jasa akan menjamin *bandwidth* dan kualitas yang akan digunakan oleh pengguna atau sebuah aplikasi. Alokasi *bandwidth* sengaja dicadangkan oleh perangkat QoS untuk pengguna tersebut. Dengan demikian pengguna tidak akan berbagi *bandwidth* dengan pengguna lain. Servis jenis ini sangat cocok untuk memberikan kualitas yang baik pada aplikasi- aplikasi *real-time* seperti *video conference*.

2. Controlled Load Service

Dalam servis jenis ini, besarnya *bandwidth* tidak dijamin akan dicadangkan untuk para pengguna jaringan tersebut. Servis ini bekerja dengan cara menjaga agar pengguna dan aplikasi didalamnya dapat selalu mendapatkan kualitas jaringan dengan *Delay* yang rendah dan *throughput* yang tinggi meskipun jaringan dalam kondisi sibuk dan padat. Dengan demikian *bandwidth* dapat digunakan dengan efisien karena tidak terbuang percuma, namun penggunaanya masih bisa mendapatkan kualitas yang terjaga. Biasanya servis jenis ini cocok digunakan dalam jaringan

dengan banyak aplikasi berbeda didalamnya. Servis ini dapat diciptakan dengan adanya RSVP dengan dibantu oleh teknologi *Weighted Random Early Detection* (WRED) (Jonathan,2011).

3. *Differentiated service*

Model QoS ini merupakan model yang sudah lama ada dalam standarisasi QoS dari organisasi IETF. Model QoS ini bekerja dengan cara melakukan klasifikasi terlebih dahulu terhadap semua paket yang masuk kedalam sebuah jaringan. Pengklasifikasian ini dilakukan dengan cara menyisipkan sebuah informasi tambahan yang khusus untuk keperluan pengaturan QoS dalam *header* IP pada setiap paket.

Setelah paket diklasifikasikan pada perangkat-perangkat jaringan terdekatnya, jaringan akan menggunakan klasifikasi ini untuk menentukan bagaimana traffic data ini diperlakukan, seperti misalnya perlakuan *queuing*, *shaping* dan *policing* nya. Setelah melalui semua proses tersebut, maka akan didapat sebuah aliran data yang sesuai dengan apa yang dikomitmenkan kepada penggunaanya.

Informasi untuk proses klasifikasi pada field *IP header* atau dengan kata lain proses klasifikasi pada layer 3 standar OSI ada dua jenis, yaitu *IP Precedence* dan *Differential Service Code Point* (DSCP). Informasi klasifikasi ini ditentukan dalam tiga atau enam bit pertama dari *field Type of Service* (ToS) pada *header paket* IP.

Klasifikasi ini juga dapat dibawa dalam frame layer 2 dalam *field Class of Service* (CoS) yang dibawa dalam frame ISL maupun 802.1Q. Tidak seperti *IntServ*, model QoS *DiffServ* ini tidak membutuhkan kemampuan QoS pada sisi pengguna dan aplikasi-aplikasi yang bekerja di dalamnya.

Metode ini merupakan metode yang paling banyak dan luas digunakan. Selain lebih mudah, lebih ringan dan lebih umum penggunaannya, implementasinya juga tidaklah terlalu sulit. Semua perangkat jaringan yang dapat bekerja berdasarkan standar TCP/IP bisa digunakan untuk melewatkan informasi QoS ini. Jadi yang perlu memiliki kemampuan pemrosesan QoS mungkin saja hanya sisi penerima dan pengirimnya saja. Tentu sistem ini jauh lebih fleksibel dan mudah diterapkan.

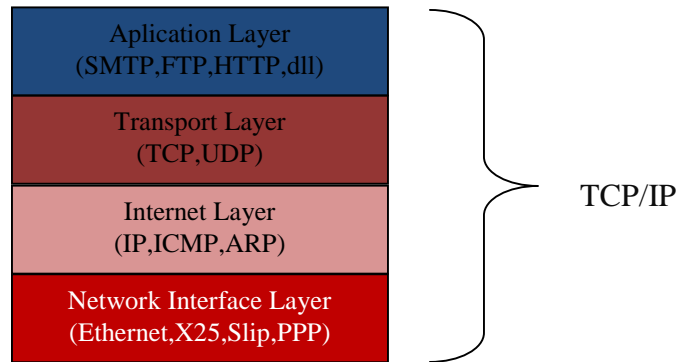
Selanjutnya pada artikel ini hanya akan dibahas teknik-teknik QoS berdasarkan sistem *DiffServ* ini (Jonathan,2011).

II.2. TCP/IP

Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme *transport* jaringan fisik yang digunakan, sehingga dapat digunakan dimana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di *Internet*.

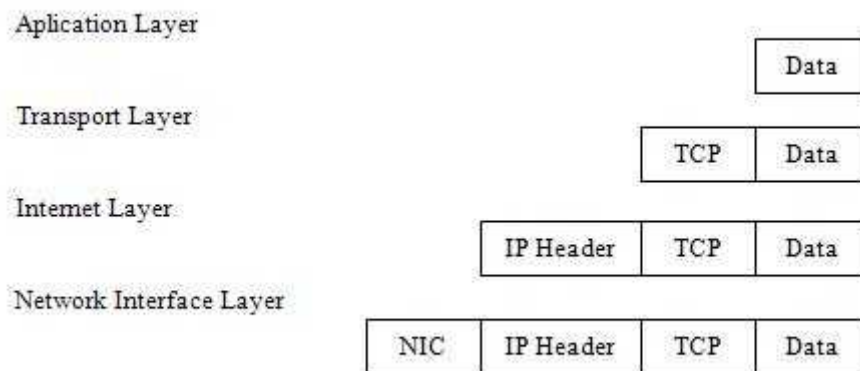
Dalam dunia komunikasi data, protokol mengatur bagaimana sebuah jaringan komputer berkomunikasi dengan komputer lain. Dalam jaringan komputer kita dapat menggunakan banyak macam protokol tapi agar sebuah komputer dapat berkomunikasi, keduanya harus menggunakan protokol yang sama. Protokol berfungsi mirip dengan bahasa. Agar dapat berkomunikasi seseorang perlu berbicara dan mengerti bahasa yang mereka gunakan.

Protokol TCP/IP dapat diterapkan dengan mudah di setiap jenis komputer dan *interface* jaringan, karena sebagian besar isi kumpulan protokol ini tidak spesifik terhadap satu komputer atau peralatan jaringan tertentu. Sekumpulan protokol TCP/IP ini dimodelkan dengan empat *Layer*, terlihat pada gambar dibawah ini :



Gambar 2.1. *Layer* TCP/IP

Berdasarkan Gambar 2.1 *Layer* TCP/IP, Jika suatu protokol menerima data dari protokol lain di layer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut, informasi ini memiliki fungsi yang sesuai dengan fungsi protokol tersebut. Setelah itu data ini diteruskan lagi ke protokol pada layer dibawahnya. Hal ini yang sebaliknya terjadi jika suatu protokol menerima data dari protokol lain yang berada pada layer dibawahnya. Jika data ini dianggap valid, protokol akan melepaskan informasi tambahan tersebut, untuk kemudian meneruskan data itu ke protokol lain yang berada pada *layer* diatasnya. Pergerakan data *layer* TCP/IP terlihat pada gambar 2.2. (Sidoarjo,2010):



Gambar 2.2. Pergerakan data dalam *layer* TCP/IP

TCP/IP terdiri atas empat *Layer* kumpulan protokol yang bertingkat seperti terlihat pada Gambar 2.1. Layer TCP/IP, *layer* tersebut adalah (Sidoarjo,2010) :

1. *Network Interface Layer*

Bertanggung jawab mengirim dan menerima data ke dan dari media fisik (berupa kabel, serat optik atau gelombang radio). Mekanisme ini harus mampu menerjemahkan sinyal listrik menjadi digital yang dimengerti komputer.

2. *Internet Layer*

Bertanggung jawab dalam proses pengiriman paket ke alamat yang tepat. Pada *layer* ini terdapat tiga macam protokol yaitu :

- a. IP (*Internet Protocol*) berfungsi untuk menyampaikan paket data ke alamat yang tepat. IP memiliki sifat sebagai *unreliable* (ketidakandalan: tidak menjamin data *datagram* yang dikirim sampai tujuan), *connectionless*, *datagram delivery services*.
- b. ARP (*Address Resolution Protocol*) protokol yang digunakan untuk menemukan alamat *hardware* dari komputer yang terletak pada *network* yang sama.
- c. ICMP (*Internet Control Message Protocol*) protokol yang digunakan untuk mengirimkan pesan dan melapor kegagalan pengiriman data.

3. *Transport Layer*

Bertanggung jawab untuk mengadakan komunikasi antara dua komputer, yang terdiri dari protokol TCP (*Transmission Control Protocol*) dan UDP (*User Datagram Protocol*). TCP merupakan protokol yang menyediakan *service connection oriented*, sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan pembentukan hubungan terlebih dahulu.

4. *Application Layer*

Terletak semua aplikasi yang menggunakan protokol TCP/IP.

II.2.1. *Internet Protocol (IP)*

Internet Protocol dikembangkan pertama kali oleh Defence Advance Research Projects Agency (DARPA) pada tahun 1970 sebagai awal dari usaha untuk mengembangkan protokol yang dapat melakukan interkoneksi berbagai jaringan komputer yang terpisah, yang masing-masing jaringan tersebut menggunakan teknologi yang berbeda. Protokol utama yang dihasilkan proyek ini adalah *Internet Protocol (IP)*. Riset yang sama dikembangkan pula yaitu beberapa protokol level tinggi yang didesain bekerja dengan IP.

Pada awal *internet* digunakan untuk menunjukkan jaringan yang menggunakan *Internet protocol (IP)* tapi dengan semakin berkembangnya jaringan, istilah ini sekarang sudah berupa istilah generik yang digunakan untuk semua kelas jaringan. *Internet* digunakan untuk menunjuk pada komunitas jaringan komputer *worldwide* yang saling dihubungkan dengan TCP/IP.

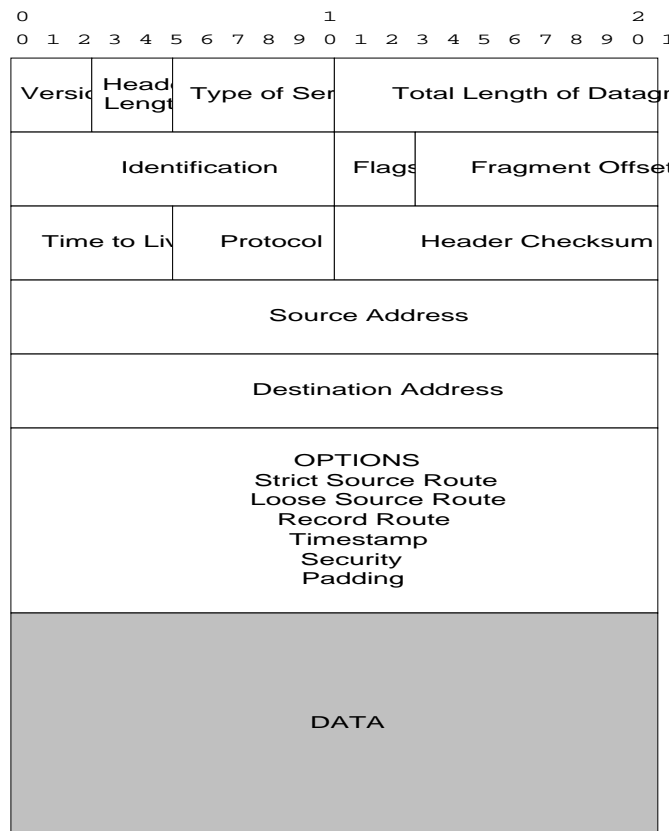
Internet Protocol didesain untuk interkoneksi sistem komunikasi komputer pada jaringan *packet switched* pada jaringan TCP/IP. Sebuah komputer diidentifikasi dengan alamat IP. Tiap komputer memiliki alamat IP yang unik, masing-masing berbeda satu sama lainnya. Hal ini dilakukan untuk mencegah kesalahan pada transfer data. Protokol data akses berhubungan langsung dengan media fisik. Secara umum protokol ini bertugas untuk menangani pendeteksian kesalahan pada saat transfer data untuk komunikasi datanya, *internet protocol* mengimplementasikan dua fungsi dasar yaitu *addressing* dan *fragmentasi*.

Salah satu hal penting dalam pengiriman informasi adalah metode pengalamatan pengirim dan penerima. Saat ini terdapat standar pengalamatan yang sudah digunakan yaitu Ipv4 dengan alamat terdiri dari 32bit. Jumlah alamat yang diciptakan dengan Ipv4 diperkirakan tidak dapat mencukupi kebutuhan pengalamatan IP sehingga sekarang sudah tersedia sistem pengalamatan yang baru yaitu yang menggunakan sistem pengalamatan 128bit. IP merupakan protokol pada *network layer* yang memiliki 2 sifat yaitu (Sidoarjo,2010):

1. **Connectionless**, yakni setiap paket data yang dikirim pada suatu saat melalui route secara independen. Paket IP akan melalui rute yang ditentukan oleh setiap *router* yang dilalui oleh *datagram* tersebut. Hal ini memungkinkan keseluruhan *datagram* tiba ditempat dalam urutan yang berbeda karena menempuh rute yang berbeda pula.
2. **Unreliable**, yakni protokol IP tidak menjamin *datagram* akan dikirim pasti sampai ke tujuan. Ia hanya akan melakukan *best-effort delivery* yakni melakukan usaha sebaik-baiknya agar paket yang dikirim tersebut sampai ke tujuan.

suatu *datagram* bisa saja tidak sampai dengan selamat ke tujuan karena beberapa hal berikut :

- a. Adanya *bit error* pada saat penransmisian *datagram* pada suatu medium.
- b. *Router* yang dilewati *discard datagram* karena terjadinya kongesti kekurangan ruang memori *buffer*.
- c. Putusnya rute ke tujuan untuk sementara waktu akibat adanya *router* yang *down*.
- d. Terjadinya kekacauan *routing*, sehingga *datagram* mengalami *looping*.



Gambar 2.3. *Datagram IP*

Berdasarkan Gambar 2.3. *Datagram IP* Setiap paket membawa data yang terdiri dari :

- Version*, yaitu versi dari protokol IP yang dipakai.
- Header Length*, berisi panjang *header* paket IP dalam hitungan 32bit word.
- Type of Service*, berisi kualitas *service* yang dapat mempengaruhi cara penanganan paket IP.
- Total Length of Datagram*, panjang IP *datagram* total dalam ukuran *byte*.
- Identification*, *Flag* dan *Fragment Offset*, berisi data yang berhubungan fragmentasi paket.
- Time to Live*, berisi jumlah *router* maksimal yang dilewati paket IP. Nilai maksimum *field* adalah 255. Setiap kali paket IP lewat satu *router*, isi dari *field* ini dikurangi satu. Jika TTL telah habis dan paket tetap belum sampai tujuan, paket ini akan dibuang dan *router* terakhir akan mengirimkan paket ICMP *time*

exceeded. Hal ini dilakukan untuk mencegah paket IP terus menerus berada pada *network*.

- g. *Protocol*, mengandung angka yang mengidentifikasi protokol *layer* atas pengguna isi data dari paket IP ini
- h. *Header Checksum*, berisi nilai *checksum* yang dihitung dari jumlah seluruh *field* dari *header* paket IP. Sebelum dikiitinkan, protokol IP tersebut untuk nantinya dihitung kembali di sisi penerima. Jika terjadi perbedaan maka paket ini dianggap rusak dan dibuang.
- i. *Source Address* dan *Destination Address*, isi dari masing-masing *field* ini cukup jelas, yakni alamat pengirim dan alamat penerima dari *datagram*. Masing-masing *field* terdiri dari 32bit, sesuai panjang IP *address* yang digunakan dalam *internet*. *Destination Address* merupakan *field* yang akan membaca oleh setiap *router* untuk menentukan kemana paket IP tersebut akan diteruskan untuk mencapai *destination address* tersebut.

II.2.2. User Datagram Protocol (UDP)

UDP yang merupakan salah satu protokol utama diatas IP dan merupakan *transport protocol* yang lebih sederhana dibanding TCP, UDP digunakan untuk situasi yang tidak mementingkan mekanisme reabilitas. *Header* UDP hanya berisi empat *field* yaitu *source port*, *destination port*, *length* dan *UDP checksum* dimana fungsinya hampir sama dengan TCP, namun fasilitas *checksum* pada UDP bersifat optional.

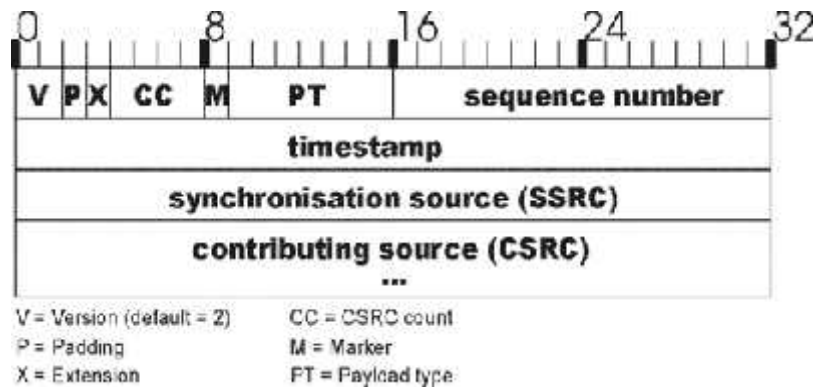
UDP pada *video* digunakan untuk mengirimkan *audio-video* secara terus menerus. UDP digunakan pada *video streaming* karena pada pengiriman nya berlangsung terus menerus dan lebih mementingkan kecepatan pengiriman data agar tiba ditujuan tanpa memperhatikan adanya paket hilang walaupun mencapai 50% dari jumlah paket yang dikirim.

Karena UDP mampu mengirimkan *data streaming* dengan cepat, maka dalam teknologi *video streaming* maupun VoIP, UDP merupakan salah satu protokol

penting yang digunakan sebagai *header* pada pengiriman data selain RTP dan IP (Sidoarjo,2010).

II.2.3. Real-Time Transport Protocol (RTP)

Protokol RTP menyediakan transfer media secara *realtime* pada paket jaringan. Protokol RTP menggunakan UDP dan *header* RTP mengandung informasi kode bit yang spesifik pada tiap paket yang dikirimkan. Hal ini membantu penerima untuk melakukan antisipasi jika terjadi paket yang hilang.



Gambar 2.4. Datagram RTP

RTP adalah protokol yang dibuat untuk mengkompensasi *jitter* dan *desequencing* yang terjadi pada jaringan IP. RTP dapat digunakan untuk beberapa informasi tipe data yang dikirim, *timestamps* yang digunakan untuk beberapa macam data *stream* yang *realtime* seperti data suara dan data video. Berdasarkan Gambar 2.4. *Datagram* RTP, Protokol RTP berisi informasi tipe data yang dikirim, *timestamps* yang digunakan untuk pengaturan waktu dan suara percakapan terdengar seperti sebagaimana diucapkan, dan *sequence number* yang digunakan untuk pengurutan paket data dan mendeteksi adanya paket hilang (Sidoarjo,2010).

RTP didesain untuk digunakan pada *transport layer*, namun demikian RTP digunakan diatas UDP, bukan pada TCP karena TCP tidak dapat beradaptasi pada pengiriman data yang *realtime* dengan keterlambatan yang relatif kecil seperti pada pengiriman data komunikasi suara. Dengan UDP dapat mengirimkan paket IP secara

multicast, RTP *stream* yang dibentuk oleh suatu terminal dapat dikirimkan ke beberapa terminal tujuan (Sidoarjo,2010).

II.3. Video Streaming

Video dapat diartikan suatu layanan menyangkut gambar bergerak atau perpindahan gambar, dan biasanya disertai dengan suara. Beberapa jenis aplikasi video (SOLPER,2011) diantaranya:

a. Broadcast Television

Yaitu program acara televisi yang disiarkan pada komputer atau monitor dengan content edukasi ataupun hiburan entertainment melalui jaringan komputer.

b. Live Event Coverage

Yaitu streaming video secara langsung atau live yang berkaitan dengan kegiatan perusahaan atau lembaga edukasi seperti pada saat event pelatihan/training di auditorium, kegiatan olahraga di stadion dan pertunjukan kesenian.

c. Surveillance

Digunakan untuk memantau suatu lokasi tertentu seperti *public safety area*, ruang produksi dan lain-lain terkait hal pengawasan dan bahkan di-*record* untuk dokumentasinya.

d. Real-Time Interactive Session

Penggunaan aplikasi video untuk *teleconference* dan *telemedicine* yang menghubungkan beberapa lokasi yang berbeda untuk berinteraksi.

e. On-Demand Video

Penerapan aplikasi video yang berdasarkan kebutuhan penggunanya dan content-nya dalam bentuk film, video tentang instruksional, dan podcasts.

II.3.1. Konsep Video Streaming

Streaming adalah sebuah teknologi untuk memutar file video atau audio secara langsung ataupun dengan *pre-recorder* dari sebuah mesin server (*web server*). Dengan kata lain, file video ataupun audio yang terletak dalam sebuah server dapat

secara langsung dijalankan pada *browser* sesaat setelah ada permintaan dari user, sehingga proses *running* aplikasi yang diunduh berupa waktu yang lama dapat dihindari tanpa harus melakukan proses penyimpanan terlebih dahulu. Saat file video atau audio di stream, akan berbentuk sebuah *buffer* di komputer *client*, dan data *audio-video* tersebut akan mulai di *download* ke dalam *buffer* yang telah terbentuk pada mesin *client*. Dalam waktu beberapa *milliseconds*, *buffer* telah terisi penuh dan secara otomatis file *video* dijalankan oleh sistem. Sistem akan membaca informasi dari *buffer* dan tetap melakukan proses *download* file, sehingga proses *streaming* tetap berlangsung ke mesin (Sidoarjo,2010).

Sebuah video yang baik tanpa dikompresi akan mengambil *bandwidth* sekitar 9Mbps, sedangkan audio yang baik tanpa dikompres akan mengambil *bandwidth* 64Kbps. Jadi total *bandwidth* yang dibutuhkan 9.064 Mbps, memang akan membutuhkan *bandwidth* yang sangat besar, namun dengan teknik kompresi yang ada, *audio-video* sebelumnya dilewatkan di jaringan TCP/IP akan terlebih dahulu melalui proses kompresi sehingga dapat menghemat kebutuhan *video* menjadi 30Kbps dan *audio* menjadi 6Kbps, artinya sebuah saluran WLAN yang memiliki *bandwidth* sebesar 54Mbps sebetulnya dapat digunakan untuk menyalurkan *audio-video* sekaligus. Tentunya untuk kebutuhan *streaming* yang *multiuser* akan dibutuhkan *multi bandwidth* pula, artinya minimal sekali kita harus menggunakan 32-36Kbps dikalikan dengan jumlah pengguna yang *streaming* dalam jaringan (Sidoarjo,2010).

II.3.2. Protocol Video Streaming

Protokol adalah aturan-aturan yang diterapkan untuk teknologi tertentu. Protokol di teknologi streaming yang digunakan untuk membawa pesan paket, dan komunikasi terjadi melalui protokol tersebut . Beberapa protokol yang digunakan dalam teknologi streaming adalah (Sidoarjo,2010):

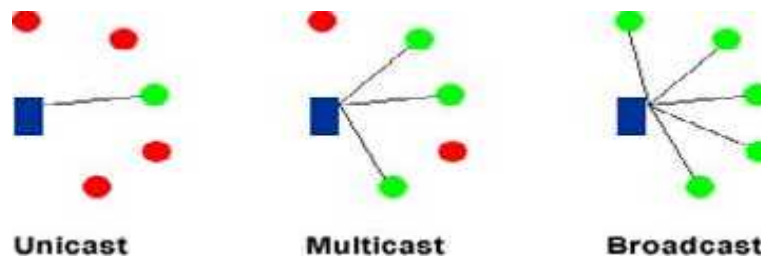
- a. *Session Description Protocol (SDP)* : Gambaran format media yang digunakan untuk menggambarkan session multimedia untuk tujuan pengumuman session, session undangan, dan bentuk-bentuk inisiasi session multimedia.
- b. *RealTime Transport Protocol (RTP)* : Sebuah paket dengan format UDP dan seperangkat konvensi yang menyediakan fungsi jaringan transportasi end-to-end, cocok untuk aplikasi transmisi data *real-time* seperti *audio*, *video* atau data simulasi, melalui layanan jaringan multicast atau unicast.
- c. *Real-time Control Protocol (RTCP)* : RTCP adalah protokol kontrol yang bekerja sama dengan RTP. Paket kontrol RTCP secara berkala dikirimkan oleh masing-masing paket dalam sesi RTP untuk semua paket lainnya. RTCP digunakan untuk mengontrol kinerja dan untuk tujuan diagnostik.
- d. *Hypertext Transfer Protocol (HTTP)* : Sebuah protokol level aplikasi yang terdistribusi, kolaboratif, dengan system informasi hypermedia. Ini adalah protokol berorientasi objek yang dapat digunakan untuk banyak tugas, seperti server nama dan sistem manajemen objek terdistribusi, melalui perpanjangan metode permintaannya.
- e. *Real Time Streaming Protocol (RTSP)* : Sebuah protokol level aplikasi untuk kontrol atas pengiriman data dengan sifat *real-time*. RTSP menyediakan kerangka extensible untuk mengaktifkan kendali pada pengiriman data *real-time*, seperti *audio* dan *video*, dengan menggunakan *Transmission Control Protocol (TCP)* atau *User Data Protocol (UDP)*.

II.3.3. Multicast transport pada aplikasi video

Aplikasi video sebaiknya menggunakan *single multicast stream* pada kliennya dibandingkan koneksi *multiple unicast*. Tujuannya adalah untuk menghemat *bandwidth* dan mengurangi beban pada server video. Sebagai gambaran, sebuah video multicast membutuhkan 3 Mbps dan ada 10 klien yang sedang streaming, maka

total *bandwidth* yang dibutuhkan adalah 3 Mbps sedangkan pada unicast transport dibutuhkan sekitar 30 Mbps.

Unicast adalah istilah untuk mendefinisikan suatu proses komunikasi dimana data informasi dikirimkan dari satu titik ke titik lain. Dalam unicast hanya ada satu pengirim dan satu penerima, atau paket data yang dikirimkan dari satu sumber ke satu tujuan, terlihat pada Gambar 2.5. Transfer paket data. Transmisi unicast bekerja pada TCP, contohnya HTTP, SMTP, FTP dan Telnet serta kartu Etherne dan IP network atau pengalaman IP mendukung unicast (SOLPER,2011).



Gambar 2.5. Transfer paket data

II.3.4. Kompresi Audio

Kompresi audio diinginkan untuk bisa menghasilkan kualitas suara HiFi dan biasanya kualitas audio digital selalu dibandingkan dengan kualitas audio CD yang menggunakan sampling 44.1 kHz dan dikuantisasi 16 bit. Terdapat banyak kompresi audio yang ada saat ini, tapi yang paling terkenal adalah standarisasi dari ISO (International Standarization Organization) yang lebih sering dikenal dengan MPEG audio. MPEG (Motion Picture Expert Group) merupakan sekumpulan ahli dibidang kompresi multimedia dengan kualitas tinggi. Salah satu group dari MPEG adalah MPEG Audio yang bekerja memfokuskan diri untuk menghasilkan kompresi audio dengan kualitas tinggi tapi dengan bit rate yang rendah. Selain MPEG Audio terdapat juga tim lain yang bekerja dibagian video dan sistem integrasi. Kualitas audio CD, yang merupakan digital audio, sangatlah bagus dan tidak bisa dibedakan dengan yang aslinya (analog). Dengan sampling rate 44.1 kHz dan 16 bit maka untuk menyimpan 1 detik musik stereo dibutuhkan $44100 * 16 * 2 = 1.4$ Mbit. Jadi satu CD audio

dengan kapasitas 650 Mbyte hanya bisa menampung musik dengan durasi sekitar 4 menit sebanyak 15 lagu, dibandingkan dengan jenis musik MP3 (hasil kompresi MPEG I Layer 3) yang bisa simpan dalam CD yang sama tapi dengan jumlah 10 kali (120 – 150 lagu).

Kompresi audio atau dikenal juga sebagai pengkodean perseptual (perceptual coding) memanfaatkan keterbatasan sistem pendengaran manusia yang tidak bisa mendengarkan dua buah sinyal dengan frekuensi berdekatan. Sinyal dengan amplituda lebih besar akan menutupi (masking) sinyal dengan amplituda yang lebih kecil. Masking effect ini dieksploitasi untuk membuang informasi redundan, sinyal yang ter-masking, dan dihasilkan informasi yang lebih sedikit secara kuantitas sambil menjaga kualitas yang sangat bagus. Hasil pengujian menghasilkan bahwa pendengar yang bertelinga “emas” sekalipun tidak bisa membedakan mana hasil kompresi dan mana sinyal aslinya. Berikut pada tabel 2.4. perbandingan bit rate dari masing-masing layer pada MPEG Audio dibandingkan dengan bit rate yang digunakan untuk merekam CD audio (1.4 Mbps) untuk mode stereo (Norazizah,2006).

Tabel 2.4. Bitrate pada MPEG Audio

Stereo channel	Bit rate (kbps)	Faktor kompresi
Layer I	384	4:1
Layer II	256	6:1
Layer III	128	12:1

Pada Tabel 2.5. Teknik kompresi audio, terdapat daftar beberapa teknik kompresi audio yang sering digunakan dengan beberapa parameter yang mencerminkan kinerja dari teknik kompresi suara tersebut (Sidoarjo,2010).

Tabel 2.5. Teknik Kompresi Audio

Name	Standardized by	Description	Bitrate (kb/s)	Sampling rate(kHz)
(ADPCM)DVI	Intel,IMA	ADPCM	32	8
G.711	ITU-T	Pulse Code Modulation	64	8
G.721	ITU-T	Adaptive Pulse Code Modulation	32	8
G.722	ITU-T	7 kHz audio coding within 64 kb/s	64	16
G.723	ITU-T	Extension of Recommendation G.72 Adaptive Different Pulse Code Modulation to 24 and 40 kb/s for digital circuit multiplication equipment application	24/40	8
G.723.1	ITU-T	dual rate speech coder for multimedia communication transmitting at 5.3 and 6.3	5.6/6.3	8
G.726	ITU-T	40,32,24,16 kb/s adaptive differential pulse code	16/24/3 2/40	8

		modulation		
G.727	ITU-T	5-,4-,3-,2- bit/sample embedded adaptive differential pulse code modulation	Var.	?
G.728	ITU-T	Coding of speech 16kb/s using low <i>Delay</i> code excited linear prediction	16	8
G.729	ITU-T	Coding of speech 8kb/s using conjugate-structure algebraic code excited linear prediction	8	8
GSM 06.10	ETSI	Regular pulse excitation long term prediction	13	8

II.3.5. Kompresi Video

Pada teknik kompresi video ada dua buah standar yang umum digunakan dalam pengiriman video melalui saluran komunikasi yang sempit yaitu:

- H.261 – biasanya menggunakan kanal ISDN dengan kecepatan $p \times 64\text{Kbps}$ dimana p adalah 1,2,3,...30
- H.263 – diarahkan untuk mengirimkan gambar video berkecepatan rendah mulai dari 20-30Kbps keatas.
- H.264 – atau sering juga disebut MPEG-4/AVC hanya memiliki file size seperempat dari video yang dibuat memakai MPEG-2

Video yang cukup baik biasanya dikirim dengan kecepatan frame per second (fps) sekitar 30fps. Jika dikirimkan tanpa kompresi, sebuah video dengan 30fps akan mengambil *bandwidth* kira-kira 9Mbps, amat sangat besar untuk ukuran kanal komunikasi data. Untuk memberikan gambaran bagaimana upaya untuk penghematan *bandwidth* dan rasio yang dibutuhkan, perhatikan tabel terlampir (Sidorajo,2010).

Tabel 2.6. Kompresi Bandwidth pada Video

Keterangan	Rata-rata PSNR (dB)	Bitrate kb/s	Rasio
Original, 30 fps	-	9124	1:1
10fps, 20Kbps	38.51	22.81	133:1
10fps, 50Kbps	41.75	56.70	54:1
10fps, 100Kbps	43.98	112.09	27:1

Tampak pada tabel 2.6. Kompresi Bandwidth pada Video, sebuah pengiriman video yang asli dengan kecepatan 30fps akan menghabiskan *bandwidth* 9Mbps. Dalam pengiriman video, biasanya kita mengurangi jumlah frame yang dikirim, misalnya menjadi 10fps. Beberapa teknik kompresi digunakan mulai dari yang paling kecil hasilnya yaitu 133:1 sampai dengan yang akan membutuhkan banyak bandwidth (500Kbps) dengan rasio kompresi 6:1. Terlihat bahwa video 10fps hasil kompresi 133:1 dapat dikirimkan dalam kanal 23Kbps dengan rata-rata signal to noise ratio 38.51dB, tentunya jika kita ingin memperoleh kualitas yang lebih baik, PSNR yang lebih baik, kompresi dapat dikurangi hingga 6:1 atau lebih rendah lagi (Sidoarjo,2010).

II.4. WIRELESS

Wireless merupakan teknologi yang bertujuan untuk menggantikan kabel yang menghubungkan terminal komputer dengan jaringan, dengan begitu komputer dapat berpindah dengan bebas dan tetap dapat berkomunikasi dalam jaringan dengan kecepatan transmisi yang memadai.

II.4.1. IEEE 802.11 Standar *Wireless* LAN

IEEE (Institute of Electrical and Electronic Engineers) merupakan institusi yang melakukan diskusi, riset dan pengembangan terhadap perangkat jaringan yang kemudian menjadi standarisasi untuk digunakan sebagai perangkat jaringan (Fauziah,2010).

II.4.1.1. Standar IEEE :

802.1 = LAN/MAN Management and Media Access Control Bridges

802.2 = Logical Link Control (LLC)

802.3 = CSMA/CD (Standar untuk Ethernet Coaxial atau UTP)

802.4 = Token Bus

802.5 = Token Ring (bisa menggunakan kabel STP)

802.6 = Distributed Queue Dual Bus (DQDB) MAN

802.7 = Broadband LAN

802.8 = Fiber Optic LAN & MAN (Standar FDDI)

802.9 = *Integrated services* LAN Interface (standar ISDN)

802.10 = LAN/MAN Security (untuk VPN)

802.11 = Wireless LAN (Wi-Fi)

802.12 = Demand Priority Access Method

802.15 = Wireless PAN (Personal Area Network) > IrDA dan Bluetooth

802.16 = Broadband Wireless Access (standar untuk WiMAX)

II.4.1.2. Perkembangan dari standar 802.11 (Fauziah,2010) :

802.11 = Standar dasar WLAN yang mendukung transmisi data 1 Mbps hingga 2 Mbps.

802.11a = Standar High Speed WLAN 5GHz band yang memiliki transfer data up to 54 Mbps.

802.11b = Standar WLAN untuk 2.4GHz yang memiliki transmisi data 5,4 hingga 11 Mbps.

802.11e = Perbaikan dari QoS (Quality of Service) pada semua interface radio IEEE WLAN.

802.11f = Mendefinisikan komunikasi inter-access point untuk memfasilitasi vendor yang mendistribusikan WLAN.

802.11g = Menetapkan teknik modulasi tambahan untuk 2,4 GHz band, untuk kecepatan transfer data hingga 54 Mbps.

802.11h = Mendefinisikan pengaturan spectrum 5 GHz band yang digunakan di Eropa dan Asia Pasifik.

802.11i = Menyediakan keamanan yang lebih baik. Penentuan alamat untuk mengantisipasi kelemahan keamanan pada protokol autentifikasi dan enkripsi.

802.11j = Penambahan pengalamatan pada channel 4,9 GHz hingga 5 GHz untuk standar 802,11a di Jepang.

II.4.1.3. Kelebihan standar 802.11 (Fauziah,2010) :

- a. Mobilitas
- b. Sesuai dengan jaringan IP
- c. Konektifitas data dengan kecepatan tinggi
- d. Frekuensi yang tidak terlisensi
- e. Aspek keamanan yang tinggi
- f. Instalasi mudah dan cepat
- g. Tidak rumit
- h. Sangat murah

II.4.1.4. Kelemahan standar 802.11 (Fauziah,2010) :

- a. Bandwidth yang terbatas karena dibagi-bagi berdasarkan spektrum RF untuk teknologi-teknologi lain.
- b. Kanal non-overlap yang terbatas.
- c. Efek multipath.
- d. Interferensi dengan pita frekuensi 2.4 GHz dan 5 GHz.

e. QoS yang terbatas.

f. Power control.

Teknologi Wireless LAN distandarisasi oleh IEEE dengan kode 802.11, tujuannya agar semua produk yang menggunakan standar ini dapat bekerja sama/kompatibel meskipun berasal dari vendor yang berbeda, 802.11b merupakan salah satu varian dari 802.11 yang telah populer dan menjadi pelopor di bidang jaringan komputer nirkabel menunjukkan bahwa 802.11b masih memiliki beberapa kekurangan di bidang keamanan yang memungkinkan jaringan Wireless LAN disadap dan diserang, serta kompatibilitas antar produk-produk Wi-Fi. Teknologi Wireless LAN masih akan terus berkembang, namun IEEE 802.11b akan tetap diingat sebagai standar yang pertama kali digunakan komputer untuk bertukar data tanpa menggunakan kabel (Fauziah,2010).

II.4.2. Standar IEEE 802.11 :

II.4.2.1. IEEE 802.11a

Standar 802.11a memungkinkan *bandwidth* yang lebih tinggi (54 Mbps *Throughput* maksimum, 30 Mbps dalam praktek). Standar 802.11a mengandung 8 saluran radio di pita frekuensi 5 GHz.

Standard IEEE 802.11a bekerja pada frekuensi 5GHz mengikuti standar dari UNII (Unlicensed National Information Infrastructure). Teknologi IEEE 802.11a tidak menggunakan teknologi spread-spectrum melainkan menggunakan standar frequency division multiplexing (FDM).

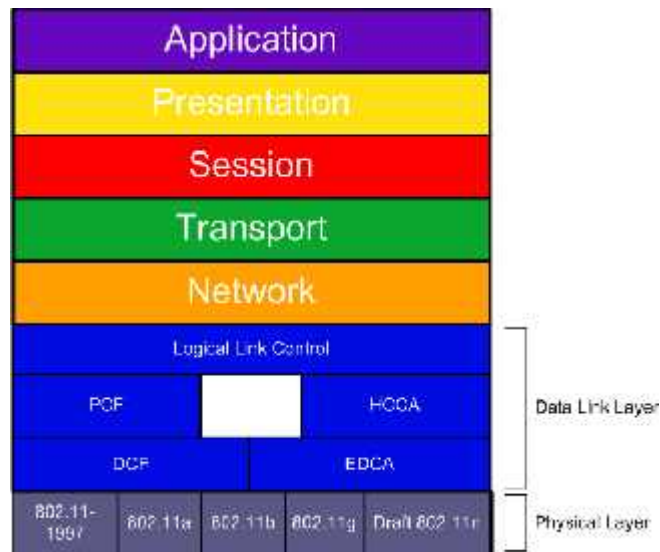
IEEE 802.11a menggunakan modulasi orthogonal frequency division multiplexing (OFDM). Regulasi FCC Amerika Serikat mengalokasikan frekuensi dengan lebar 300MHz di frekuensi 5GHz. Tepatnya 200MHz di frekuensi 5.150 - 5.350 Mhz. Dan sekitar 100MHz *bandwidth* pada frekuensi 5.725 - 5.825 Mhz (Fauziah,2010).

II.4.2.2. IEEE 802.11b

Standar 802.11b saat ini yang paling banyak digunakan satu. Menawarkan throughput maksimum dari 11 Mbps (6 Mbps dalam praktek) dan jangkauan hingga 300 meter di lingkungan terbuka. Ia menggunakan rentang frekuensi 2,4 GHz, dengan 3 saluran radio yang tersedia (Fauziah,2010).

II.4.2.3. IEEE 802.11e

Standar 802.11e yang dimaksudkan untuk meningkatkan kualitas layanan pada tingkat data link layer. Tujuan standar ini adalah untuk menentukan persyaratan paket yang berbeda dalam hal *bandwidth* dan keterlambatan transmisi sehingga memungkinkan transmisi yang lebih baik suara dan video (Fauziah,2010).



Gambar 2.6. Diagram OSI 7-layer standar 802.11e (SOLPER,2011)

IEEE 802.11e adalah sebuah amandemen dari 802.11 yang khusus membahas tentang perbaikan Quality of service pada 802.11 dengan menambahkan beberapa fungsi tertentu pada MAC layer. IEEE 802.11e mendefinisikan fungsi koordinasi baru dinamakan Hybrid Coordination Function (HCF). HCF menyediakan mekanisme akses baik secara terpusat yaitu HCF Controlled Channel Access (HCCA)

maupun secara terdistribusi yaitu Enhanced Distributed Channel Access (EDCA) (Fauziah,2010).

a. Enhanced Distributed Channel Access (EDCA)

Dirancang untuk menyediakan QoS dengan menambahkan fungsi pada DCF. Pada MAC layer, EDCA mendefinisikan empat FIFO queue yang dinamakan Access Category (AC) yang memiliki parameter EDCA tersendiri. Mekanisme aksesnya secara umum hampir sama dengan DCF, hanya saja durasi DIFS digantikan dengan AIFS. Sebelum memasuki MAC layer, setiap paket data yang diterima dari layer di atasnya di-assign dengan nilai prioritas user yang spesifik antara 0 sampai 7. Setiap paket data yang sudah diberi nilai prioritas dipetakan ke dalam Access Kategori. Nilai parameter EDCA berbeda untuk AC yang berbeda. Parameter-parameter tersebut adalah (Fauziah,2010) :

1. AIFS (Arbitration Inter-Frame Space)

Setiap AC memulai prosedur backoff atau memulai transmisi setelah satu periode waktu AIFS menggantikan DIFS.

2. CWmin, CWmax.

Nilai backoff counter merupakan nilai random terdistribusi uniform antara contention window CWmin dan CWmax.

3. TXOP (Transmission Opportunity) limit

durasi maksimum dari transmisi setelah medium diminta. TXOP yang diperoleh dari mekanisme EDCA disebut EDCA-TXOP. Selama EDCA-TXOP, sebuah station dapat mentransmisikan multiple data frame dari AC yang sama, dimana periode waktu SIFS memisahkan antara ACK dan transmisi data yang berurutan. TXOP untuk setiap AC ke-i didefinisikan sebagai $TXOP[i] = (MSDU[i]/R) + ACK + SIFS + AIFS[i]$, MSDU [i] adalah panjang paket pada AC ke-i. R adalah rate transmisi physical, ACK adalah waktu yang dibutuhkan untuk mentransmisikan ack, SIFS adalah periode waktu SIFS, AIFS[i] adalah waktu AIFS pada AC ke-i (Fauziah,2010).

b. HCF Controlled Channel Access (HCCA)

Menyediakan akses ke medium secara polling. HC menggunakan PCF Interframe Space (PIFS) untuk mengontrol kanal kemudian mengalokasikan TXOP pada station. Polling dapat berada pada periode contention (CP), dan penjadwalan paket dilakukan berdasarkan Traffic Specification (TSPEC) yang diperbolehkan.

c. Fuzzy Logic

Metode ini sudah banyak dipakai pada sistem kontrol karena sederhana, cepat dan adaptif. Sistem Inferensi Fuzzy (FIS) adalah sistem yang dapat melakukan penalaran dengan prinsip serupa seperti manusia melakukan penalaran dengan nalurinya. FIS tersebut bekerja berdasarkan kaidah-kaidah linguistik dan memiliki algoritma fuzzy yang menyediakan sebuah aproksimasi untuk dimasuki 3 analisa matematika (Fauziah,2010).

Untuk memperoleh output, diperlukan 3 tahapan yaitu (Fauziah,2010) :

1. Fuzzification

merupakan suatu proses untuk mengubah suatu peubah masukan dari bentuk tegas (crisp) menjadi peubah fuzzy (variabel linguistik) yang biasanya disajikan dalam bentuk himpunan-himpunan fuzzy dengan fungsi keanggotaannya masing-masing.

2. Rule evaluation (Evaluasi aturan)

merupakan proses pengambilan keputusan (inference) yang berdasarkan aturan-aturan yang ditetapkan pada basis aturan (rules base) untuk menghubungkan antar peubah-peubah fuzzy masukan dan peubah fuzzy keluaran.

3. Defuzzification

Input dari proses defuzzifikasi adalah suatu himpunan fuzzy yang diperoleh dari komposisi aturan-aturan fuzzy, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan fuzzy tersebut. Jika diberikan suatu himpunan fuzzy dalam range tertentu, maka harus dapat di ambil suatu nilai crisp tertentu sebagai output.

II.4.2.4. IEEE 802.11g

Standar 802.11g menawarkan *bandwidth* yang tinggi (54 Mbps *Throughput* maksimum, 30 Mbps dalam praktek) pada rentang frekuensi 2,4 GHz. Standar 802.11g mundur-kompatibel dengan standar 802.11b, yang berarti bahwa perangkat yang mendukung standar 802.11g juga dapat bekerja dengan 802.11b.

Dalam evolusi WLAN adalah pengenalan IEEE 802.11g. Ini merupakan standar IEEE 802.11g akan secara dramatis dapat meningkatkan performa WLAN. IEEE 802.11g adalah sebuah standar jaringan nirkabel yang bekerja pada frekuensi 2,45 GHz dan menggunakan metode modulasi OFDM. 802.11g yang dipublikasikan pada bulan Juni 2003 mampu mencapai kecepatan hingga 54 Mb/s pada pita frekuensi 2,45 GHz, sama seperti halnya IEEE 802.11 biasa dan IEEE 802.11b. Standar ini menggunakan modulasi sinyal OFDM, sehingga lebih resistan terhadap interferensi dari gelombang lainnya (Fauziah,2010).

II.4.2.5. IEEE 802.11n

IEEE 802.11n merupakan salah satu standarisasi yang sudah direvisi dari versi sebelumnya IEEE 802.11-2.007 sebagaimana telah dirubah dengan IEEE 802.11k-2008, IEEE 802.11r-2008, IEEE 802.11y-2008, dan IEEE 802.11w-2009, dan didasarkan pada standar IEEE 802.11 sebelumnya dengan menambahkan Multiple-Input Multiple-Output (MIMO) dan 40 MHz saluran ke layer fisik, dan frame agregasi ke MAC layer (Fauziah,2010).

Tabel 2.7. Spesifikasi protokol 802.11

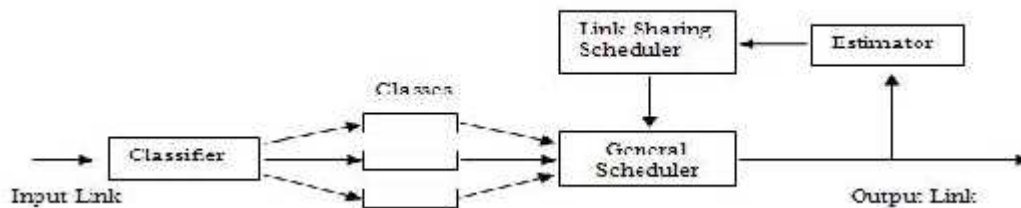
802.11 Protocol	Frekuensi (Ghz)	<i>Bandwidth</i> (Mhz)	Jarak maksimal dalam ruangan (m)	Jarak maksimal luar ruangan (m)
A	3.7 – 5	20	20	100
B	2.4	20	35	140
G	2.4	20	38	140
N	2.4 - 5	20 - 40	70	250

II.5. HTB (*HIERARCHICAL TOKEN BUCKET*)

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang baru-baru ini diperkenalkan bagi *router* berbasis Linux, dikembangkan pertama kali oleh Martin Devera pada akhir 2001 untuk diproyeksikan sebagai pilihan (atau pengganti) mekanisme penjadwalan yang saat ini masih banyak dipakai yaitu CBQ. HTB diklaim menawarkan kemudahan pemakaian dengan teknik peminjaman dan implementasi pembagian trafik yang lebih akurat (Pangera,2004).

Pada HTB terdapat parameter *ceil* sehingga kelas akan selalu mendapatkan *bandwidth* di antara base link dan nilai *ceil* linknya. Parameter ini dapat dianggap sebagai Estimator kedua, sehingga setiap kelas dapat meminjam *bandwidth* selama *bandwidth* total yang diperoleh memiliki nilai di bawah nilai *ceil*. Hal ini mudah diimplementasikan dengan cara tidak mengijinkan proses peminjaman *bandwidth* pada saat kelas telah melampaui link ini (keduanya leaves dan interior dapat memiliki *ceil*). Sebagai catatan, apabila nilai *ceil* sama dengan nilai base link, maka akan memiliki fungsi yang sama seperti parameter bounded pada CBQ, di mana kelas-kelas tidak diijinkan untuk meminjam *bandwidth*. Sedangkan jika nilai *ceil* diset tak terbatas atau dengan nilai yang lebih tinggi seperti kecepatan link yang dimiliki, maka akan didapat fungsi yang sama seperti kelas non-bounded (Pangera,2004).

Cara kerja HTB tidak ada perbedaan dengan pendahulunya yaitu CBQ, hanya saja pada General Scheduler pada HTB menggunakan mekanisme Deficit Round Robin (DRR) *flowchart* terlihat pada Gambar 2.7. Deficit Round robin dan pada blok umpan baliknya, estimator HTB menggunakan *Token Bucket Filter* (TBF)



Gambar 2.7. Deficit Round Robin (Arifin,2012)

II.5.1. *General Schedule* HTB

HTB menganggap hirarki kelas lengkap dan trafik dipisah-pisah menjadi beberapa aliran trafik, algoritma untuk penjadwalan adalah sebagai berikut :

- Pertama memilih kelas pada cabang terendah (*leaf class*) yang link nya belum mencapai batas.
- Kemudian mulai mengirimkan paket dari kelas yang memiliki prioritas tertinggi kemudian berlanjut ke yang rendah, apabila link semua kelas melampaui batas *link* maka dilakukan tes melalui suatu putaran lengkap untuk menemukan *leaf class* yang tidak dapat meminjam *bandwidth* dari kelas di atasnya (*parent class*) jika tidak ada maka putaran diulangi dengan mencoba meminjam *bandwidth* dari kelas diatas *parent class* (*grandfather class*) (Arifin,2012).

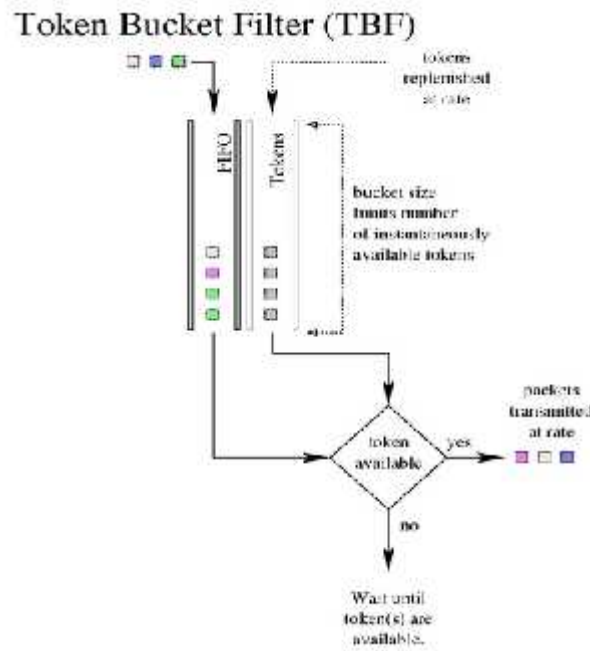
II.5.2. *Estimator*

Hierarchical Token Bucket (HTB) menggunakan *Token Bucket Filter* (TBF) sebagai *estimator* untuk menentukan apakah suatu kelas berada dalam keadaan *underlimit*, *atlimit*, atau *overlimit*. TBF bekerja dengan dasar algoritma ember token, setiap paket yang akan dikirimkan harus memiliki token yang berada dalam ember token, jika token tidak tersedia didalam ember maka paket yang dikirimkan harus menunggu sampai tersedia token yang cukup untuk mengirimkan paket yang sedang menunggu (Arifin,2012).

Teknik antrian HTB memberikan kita fasilitas pembatasan trafik pada setiap level maupun klasifikasi, bandwidth yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah. Kita juga dapat melihat HTB seperti suatu struktur organisasi dimana pada setiap bagian memiliki wewenang dan mampu membantu bagian lain yang memerlukan (Santosa,2007).

Token bucket filter (TBF) membatasi bandwidth dengan metode *shape & drop*, prinsip kerja menggunakan aliran token yang memasuki bucket dengan kecepatan (rate) konstan, jika token dalam bucket habis maka paket data akan di antri dan kelebihananya dibuang, setiap paket data yang dikeluarkan identik dengan token.

Token dalam bucket akan lebih cepat habis jika aliran paket data melampaui kecepatan token memasuki bucket, jadi kita asumsikan bahwa trafik melebihi batas konfigurasi (Santosa,2007).



Gambar 2.8. Token Bucket Filter

Implementasi TBF terdiri dari sebuah *Buffer (bucket)* terlihat pada Gambar 2.8. Token Bucket Filter, yang secara konstan diisi oleh beberapa informasi virtual yang dinamakan token. Parameter paling penting dari *bucket* adalah ukuran nya, yaitu banyaknya token yang dapat disimpan. Setiap token yang masuk mengumpulkan satu paket yang datang dari antrian data dan kemudian dihapus dari *bucket*. Dengan menghubungkan algoritma ini dengan dua aliran token dan data akan didapati tiga buah kemungkinan skenario :

1. Data yang datang pada TBF memiliki *link* yang sama dengan masuknya token. Dalam hal ini, setiap paket yang masuk memiliki token masing-masing dan akan melewati antrian tanpa adanya *Delay*.
2. Data yang datang pada TBF memiliki *link* yang lebih kecil daripada *link token*. Hanya sebagian token yang dihapus pada output, memenuhi ukuran bucket.

Token yang tidak digunakan kemudian akan dapat digunakan untuk mengirimkan data pada kecepatan yang melampaui *link token* standar, ini terjadi jika ada ledakan data yang pendek.

3. Data yang datang pada TBF memiliki *link* yang lebih besar daripada *link token*. Hal ini berarti *bucket* akan segera kosong dari token, yang menyebabkan TBF akan menutup alirannya untuk sementara. Hal ini lah yang dinamakan situasi *overlimit*. Jika paket tetap datang, maka paket akan segera dibuang (Arifin,2012).

Parameter TBF (Santosa,2007):

- a. rate

batas bandwidth yang di set oleh administrator, jika aliran paket data melebihi nilai ini maka data akan di buang (drop) atau mengalami penundaan, bandwidth dipotong.

- b. Limit / latency

limit merupakan jumlah byte yang dapat diantri sebelum token tersedia, sedangkan latency adalah lama waktu (dalam mili detik [msec]) paket dapat diantri.

- c. Burst/buffer/maxburst

Kapasitas bucket dalam byte, paket data yang melebihi nilai ini akan dibuang atau mengalami penundaan.

- d. Peakrate

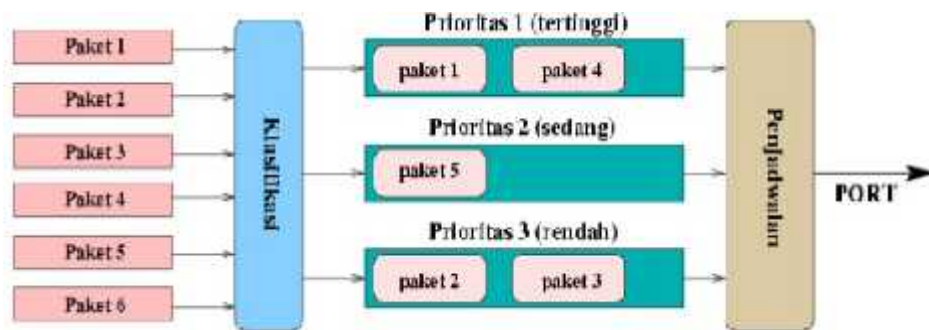
Batas maksimum rate menangani lonjakan bandwitdh sesaat dengan syarat paket data tidak boleh melebihi kapasitas bucket dan mtu.

II.5.3. Metode pengendalian trafik

Dalam mengendalikan trafik administrator jaringan bisa memilih beberapa metode tergantung dari situasi pada jaringan LAN atau backbone. Tiap trafik akan dikendalikan dengan metode tertentu yang akan berdampak pada kecepatan akses, jadi administrator jaringan perlu membaca dan mengerti bagian ini terlebih dahulu, beberapa metode pengendalian trafik sebagai berikut:

1. Prioritas

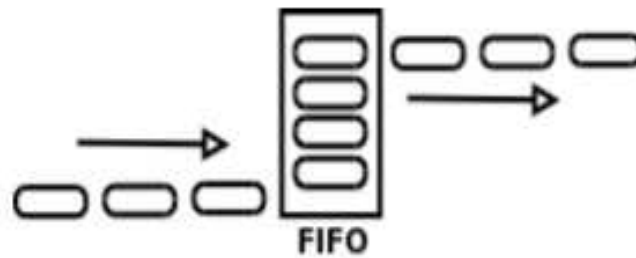
Pada metode prioritas paket data yang melintasi gateway diberikan prioritas berdasarkan port, alamat IP atau sub net. Jika trafik pada gateway sedang tinggi maka prioritas dengan nilai terendah (nilai paling rendah berarti prioritas tertinggi) akan di proses terlebih dahulu, sedangkan yang lainnya akan di berikan ke antrian atau dibuang seperti terlihat pada Gambar 2.9. konsep prioritas antrian. Metode prioritas paling cocok diterapkan pada koneksi internet yang memiliki bandwidth sempit, hanya trafik paling penting saja yang dilewatkan seperti smtp dan pop3 (Santosa,2007).



Gambar 2.9. Prioritas Antrian

2. FIFO

Pada metode FIFO jika trafik melebihi nilai set maka paket data akan dimasukkan ke antrian, paket data tidak mengalami pembuangan hanya tertunda beberapa saat seperti terlihat pada Gambar 2.10 model FIFO. Metode FIFO cocok diterapkan pada koneksi internet dengan bandwidth menengah 64kbps, untuk menghindari bootle neck pada jaringan LAN. Paket data jika melebihi batas konfigurasi akan di masukkan ke dalam antrian dan pada saat jaringan LAN tidak sibuk maka paket data dalam antrian akan dikeluarkan (Santosa,2007).



Gambar 2.10. Antrian FIFO

3. Penjadwalan

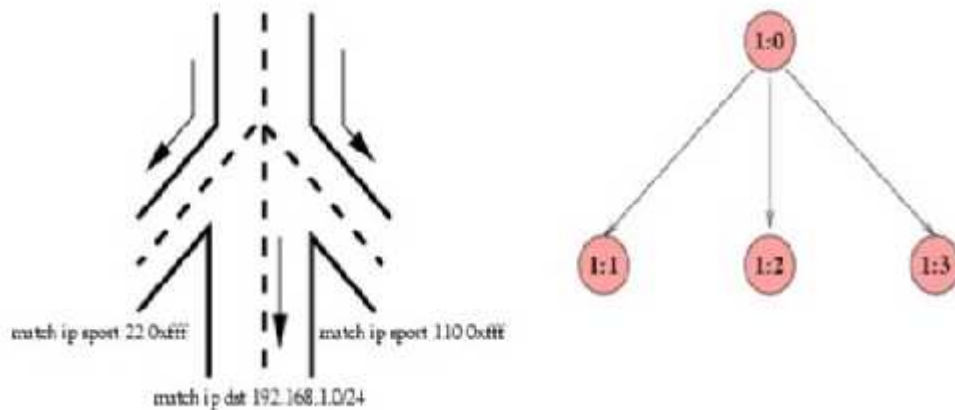
Metode penjadwalan atau scheduling ini paling sering dipakai karena memiliki kemampuan membagi paket data ke dalam ukuran yang sama besar kemudian memasukkan ke dalam beberapa antrian. Antrian itu kemudian di keluarkan oleh scheduler dengan algoritma round robin (Santosa,2007).

4. Shape & drop

Shape & drop merupakan metode paling cocok serta efektif untuk jaringan yang memiliki beban trafik sangat tinggi. Jika trafik melebihi nilai set maka paket data akan di masukan ke dalam antrian sehingga trafik menurun secara perlahan, metode ini disebut pemotongan bandwidth, kemudian jika trafik terus menerus melebihi nilai set maka paket data akan dibuang (drop) (Santosa,2007).

5. Filtering

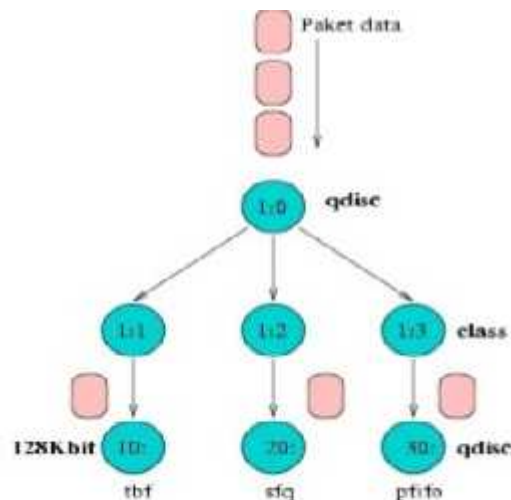
Paket data yang akan memasuki antrian melalui klasifikasi (class) sebelumnya disaring (filter) terlebih dahulu, agar setiap paket bisa ditentukan jenisnya, alamat IP, alamat PORT dan TOS nya terlihat pada Gambar 2.11. filtering paket. Kemudian fungsi yang kedua, filtering digunakan untuk mengarahkan suatu paket agar ke tujuan yang benar, ke klasifikasi paket (class) sesuai dengan arah alirannya (Santosa,2007).



Gambar 2.11. Filtering Paket

6. Klasifikasi Paket

Klasifikasi paket merupakan cara memberikan suatu kelas atau perbedaan pada setiap paket, Gambar 2.12. klasifikasi paket data. Hal ini dilakukan untuk mempermudah penanganan paket oleh antrian. Klasifikasi berbeda dengan filtering yang berfungsi mengarahkan dan menyaring aliran paket data. Contoh pada gambar XII. dibawah ini menunjukkan paket data dibagi menjadi tiga kelas 1:1, 1:2 dan 1:3 dan tiap kelas tersebut ditangani oleh teknik antrian (qdisc) 10: (tbf), 20:(sfq) dan 30: (pfifo) (Santosa,2007).



Gambar 2.12. Klasifikasi paket data

II.6. ACTION RESEARCH

Action Research atau penelitian tindakan merupakan salah satu bentuk rancangan penelitian, dalam penelitian tindakan peneliti mendeskripsikan, menginterpretasi dan menjelaskan suatu situasi sosial pada waktu yang bersamaan dengan melakukan perubahan atau *intervensi* dengan tujuan atau partisipasi. *Action Research* dalam pandangan tradisional adalah suatu kerangka penelitian pemecahan masalah, dimana terjadi kolaborasi antara peneliti dan *client* dalam mencapai tujuan (Kurt Lewin, 1973 disitasi Sulaksana, 2004), sedangkan pendapat Davison, Martinsons & Kock (2004), menyebutkan penelitian tindakan, sebagai sebuah metode penelitian, didirikan atas asumsi bahwa teori dan praktik dapat secara tertutup diintegrasikan dengan pembelajaran dari hasil survei *intervensi* yang direncanakan setelah *diagnosis* yang rinci terhadap konteks masalahnya.

II.6.1. Tujuan dan Ciri-ciri Penelitian Tindakan

Penelitian tindakan bertujuan untuk memperoleh pengetahuan untuk situasi atau sasaran khusus dari pada pengetahuan yang secara ilmiah tergeneralisasi. Pada umumnya penelitian tindakan untuk mencapai tiga hal berikut :

1. Peningkatan Praktek.
2. Peningkatan (Pengembangan Profesional) pemahaman praktik dan praktisinya.
3. Peningkatan situasi tempat pelaksanaan praktek.

Hubungan antara peneliti dan hasil penelitian tindakan dapat dikatakan hasil penelitian tindakan dipaia sendiri oleh penelitiannya dan tentusaja oleh orang lain yang menginginkannya dan penelitian terjadi di dalam situasi nyata yang pemecahan masalanya segera diperlukan, dan hasilnya langsung diterapkan dalam situasi terkait. Selain itu, tampak bahwa dalam penelitian tindakan peneliti melakukan pengolahan, penelitian, dan sekaligus pengembangan.

Penelitian tindakan dilaksanakan bersama-sama paling sedikit dua orang yaitu antara peneliti dan partisipan atau klien yang berasal dari akademis ataupun masyarakat. Oleh karna itu, tujuan yang akan diacapai dari suatu penelitian tindakan

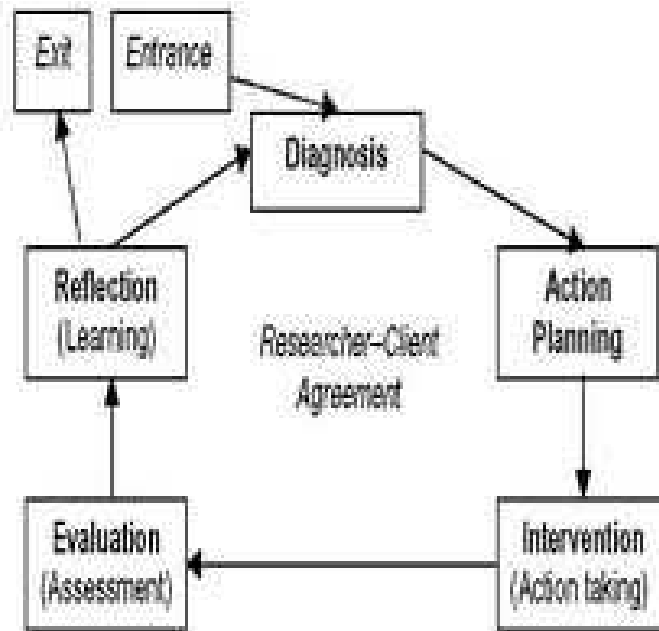
akan dicapai dan berakhir tidak hanya pada situasi organisatoris tertentu, melainkan terus dikembangkan berupa aplikasi atau teori kemudian hasilnya akan di publikasikan ke masyarakat dengan tujuan riset.

Dilihat dari ruang lingkup, tujuan, metode, dan prakteknya, action research dapat dianggap sebagai penelitian ilmiah micro. Action research adalah penelitian yang bersifat partisipatif dan kolaboratif. Maksudnya, penelitiannya dilakukan sendiri oleh peneliti, dan diamati bersama dengan rekan-rekannya. Action research berbeda dengan studi kasus karena tujuan dan sifat kasusnya yang tidak unik seperti pada studi kasus, action research tidak digunakan untuk menguji teori. Namun kedua macam penelitian ini mempunyai kesamaan, yaitu bahwa peneliti tidak berharap hasil penelitiannya akan dapat digeneralisasi atau berlaku secara umum (Zahroh,2013).

Dalam bidang pendidikan, action research dianggap sebagai alternatif dari penelitian tradisional (penelitian yang biasa dilakukan). Modal utama peneliti action research adalah pengalamannya dalam bidang yang digeluti dan pengetahuan yang ia miliki. Sebenarnya action research dapat juga dilakukan dalam skala besar karena seperti dikatakan di atas, action research dilakukan bersama rekan-rekan seprofesi, sehingga mereka dapat berbagai pengalaman untuk kepentingan mereka masing-masing. Action research merupakan metode yang handal untuk menjembatani teori dan praktek (dalam pendidikan), karena dengan action research para guru dianjurkan menemukan dan mengembangkan teorinya sendiri dari prakteknya sendiri (Zahroh,2013).

Berdasarkan Gambar 2.13. Siklus *Action Research*, berikut tahapan penelitian tindakan yang dapat ditempuh menurut Davison, Martin & Kock (2004) :

1. Melakukan Diagnosa (*Diagnosing*)
2. Membuat Rencana Tindakan (*Action Planning*)
3. Melakukan tindakan (*Action Taking*)
4. Melakukan Evaluasi (*evaluating*)
5. Pembelajaran (*Learning*)

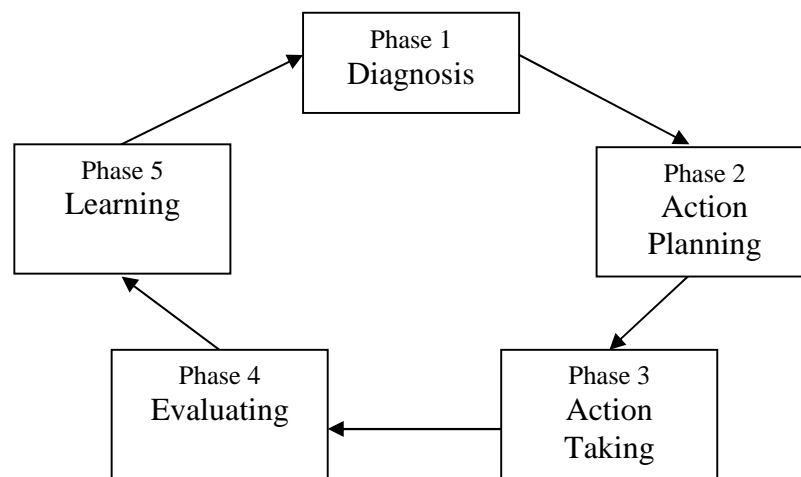


Gambar 2.13. Siklus *Action Research*

BAB III

METODOLOGI PENELITIAN

Metodologi penelitian menjelaskan rangkaian proses penelitian yang dilakukan dalam menganalisa *Quality Of Service* dari jaringan *Wireless*. Adapun metodologi yang digunakan adalah *Action research*. Davison, Martinsons & Kock (2004), membagi *Action research* dalam 5 tahapan yang merupakan siklus, yaitu:



Gambar 3.1 *Flowchart* Metodologi Penelitian

III.1. *Diagnosis* (Melakukan diagnosa)

Melakukan identifikasi masalah-masalah pokok yang ada guna menjadi dasar penelitian sehingga terjadi perubahan, analisa *Quality of Service video streaming* pada jaringan *Wireless*, pada tahap ini peneliti mengidentifikasi kebutuhan pengguna terhadap *video streaming*. Ditempuh dengan cara melihat dari hasil survey yang telah dilakukan oleh beberapa media tentang perkembangan internet dan apa saja yang paling banyak diakses oleh pengguna dengan perangkat apa saja pengguna mengakses konten dari *internet*. Terkait dengan itu, kebutuhan akan *video streaming* membutuhkan *bandwidth* yang lumayan besar apalagi dengan tingkat *bitrate video* yang

berbeda-beda, semakin tinggi bitrate semakin banyak *bandwidth* yang dibutuhkan. Jarak pengguna terhadap *wireless* juga sangat berpengaruh, dikarenakan *throughput* yang didapat pengguna akan berbeda. (Zahroh,2013)

III.2. Action Planning (Rencana Tindakan)

Peneliti memahami pokok masalah yang ada kemudian dilanjutkan dengan menyusun rencana tindakan yang tepat untuk menyelesaikan masalah yang ada, pada tahap ini peneliti memasuki tahap analisa kebutuhan untuk penelitian, baik itu dari segi perangkat apa saja yang akan di gunakan, kebutuhan akan mengakses *video streaming* dan data teks hingga materi penunjang untuk mendukung implementasi dari penelitian. Dengan memperhatikan kebutuhan ini peneliti akan membuat rencana awal untuk melakukan penelitian ini. Pada tahap ini peneliti akan merancang sebuah skema jaringan dan membangun server berbasis *unix* yang akan digunakan untuk memanajemen *bandwidth* nantinya pada tahap implementasi. (Zahroh,2013)

III.3. Action Taking (Melakukan Tindakan)

Peneliti mengimplementasikan rencana tindakan dengan harapan dapat menyelesaikan masalah. Selanjutnya setelah model dibuat berdasarkan rancangan, maka akan dilakukan tahap implementasi yaitu akan dilakukan beberapa analisa dari komponen dari jaringan yaitu *Delay*, *jitter*, *bandwidth*, dan *packet loss* ketika dibebani oleh *video streaming* dengan kualitas video tertentu dan dengan *datarate* tertentu. Akan dilakukan juga perbandingan menggunakan HTB dan tanpa HTB. Menggunakan *Tools Wireshark* untuk mendapatkan hasil dari tiap atribut dari Qos. (Zahroh,2013)

III.4. Evaluating (Melakukan Evaluasi)

Setelah masa implementasi (action taking) dianggap cukup kemudian peneliti melaksanakan evaluasi hasil dari implementasi tadi, dalam tahap ini membuat suatu kesimpulan dari beberapa hasil yang didapat dari atribut yang di analisa pada tahap sebelumnya. (Zahroh,2013)

III.5. *Learning* (Pembelajaran)

Tahap ini merupakan bagian akhir siklus yang telah dilalui dengan melaksanakan review tahap-pertahap yang telah berakhir kemudian penelitian ini dapat berakhir. Beberapa atribut dari Qos yang telah didapat nilai nya akan di *review* dan akan didapat suatu kesimpulan yang akan menjadi standar untuk penelitian ini. Dan hasil penelitian ini juga dapat dipertimbangkan untuk tindakan kedepan yang bisa dilakukan dalam kaitan dengan penelitian ini. (Zahroh,2013)

BAB IV

ANALISA DAN PERANCANGAN

Pada bab ini, penulis akan menjelaskan proses penerapan *Quality of Service* (QoS) *Video Streaming* pada jaringan *Wireless*. Dalam hal ini yang menjadi fokus adalah mekanisme pengukuran *Quality of Service* (QoS) pada *Video Streaming* dengan menggunakan jaringan *Wireless* dalam lingkungan jaringan area lokal (LAN) sederhana. Konsep QoS yang penulis gunakan adalah dengan membagi satu jaringan utama menjadi beberapa kelas yang memiliki layanan jaringan yang berbeda, hal ini dimaksudkan untuk memberikan layanan kepada *user* dengan tingkat prioritas dan kebutuhan akan layanan yang berbeda sehingga setiap *user* mendapatkan layanan sesuai dengan kebutuhan dan prioritas serta penggunaan sumber daya jaringan dapat lebih optimal.

IV.1. ANALISA KEBUTUHAN SISTEM

IV.1.1. Diagnosis (Melakukan Diagnosa)

Perkembangan internet di indonesia semakin pesat, hal yang sangat berpengaruh terhadap perkembangan penggunaan *internet* yaitu penggunaan *smartphone* yang meningkat, *netbook*, *tablet*, *notebook*, perangkat selular dan juga penggunaan modem GSM/CDMA. Sebanyak 40% dari pengguna Internet di Indonesia (24,2 juta orang) mengakses Internet lebih dari 3 jam setiap harinya dengan mayoritas pengguna berada pada rentang usia 15-35 tahun, menurut survey MarkPlus Insight (dailysocial.net pada tanggal 13/11/2012). Pengguna *internet* di indonesia paling banyak mengakses situs musik, *film*, *video games* dan *live streaming*. Kebanyakan dari pengguna internet ini mengunjungi situs-situs yang menyediakan

informasi yang berhubungan dengan minat/hobi mereka, hasil survey Tekno-Kompas (tekno.kompas.com pada tanggal 17/04/2012).

Meningkatnya penggunaan *internet* yang di dasari oleh perkembangan penggunaan *smartphone* di indonesia dengan tingkat mengakses konten terbanyak yaitu *streaming* musik, video dan juga *games online*. Perkembangan teknologi *wireless* juga sudah menjamur di berbagai bidang, seperti yang telah kita ketahui terdapat banyak titik *Hotspot* di berbagai tempat mulai dari yang gratisan hingga berbayar. namun Video *Streaming* menggunakan jaringan *wireless* berbeda dengan kabel karena karakteristik *wireless* yang terbatas dibanding kabel. Jaringan *wireless* bersifat distance sensitive, yaitu performansi akan menurun jika pengguna semakin jauh dari Access Point. Tapi itu akan tertutupi jika ada penanganan yang baik. Terkait dengan itu, kebutuhan akan *video streaming* membutuhkan *bandwidth* yang lumayan besar apalagi dengan tingkat *bitrate video* yang berbeda-beda, semakin tinggi *bitrate* semakin banyak *bandwidth* yang dibutuhkan. Maka dari itu dibutuhkan penanganan *Quality Of Service* (QOS) pada implementasi Video *Streaming* menggunakan jaringan *Wireless* ini. Karena jaringan *Wireless* juga merupakan jaringan berbasis IP maka secara default memiliki QoS best effort yang tidak membedakan paket yang dikirim apakah itu file *voice*, *video*, atau *data teks*. Jadi ketika file tersebut dilewatkan dalam jaringan ada kemungkinan terjadi *jitter* dan *packet loss* karena tidak adanya kontrol kualitas layanan (*Quality of Service*) yang juga terkait ketersediaan *bandwidth* yang rendah dari kapasitas yang diperlukan. Keadaan dapat mempengaruhi kualitas file yang dikirimkan atau bahkan dapat terjadi kerusakan pada file sesampainya di tujuan. Maka dari itu diperlukan efisiensi *bandwidth* dengan mengaplikasikan konsep kontrol kualitas layanan (*Quality of Service*).

IV.1.2. Action Planning (Rencana Tindakan)

Peneliti memahami pokok masalah yang ada kemudian dilanjutkan dengan menyusun rencana tindakan yang tepat untuk menyelesaikan masalah yang ada, pada tahap ini peneliti memasuki tahap analisa kebutuhan untuk penelitian.

IV.1.2.1. Persiapan Perangkat

Pada tahap ini peneliti telah memahami sebelumnya masalah yang akan diteliti dan peneliti mempersiapkan peralatan yang diperlukan. Adapun Perangkat yang digunakan pada penelitian ini berupa :

1. Spesifikasi *Hardware*

a) Spesifikasi *Server Streaming*

Server yang digunakan sebagai tempat sharing video yaitu sebuah *Personal Computer* (PC) dengan spesifikasi sebagai berikut :

Motherboard : Asus P5

Processor : AMD ATHLON II X2 260

Memory : 4GB

Harddisk : 80GB

Lan card : TP-Link 10/100 Mbps

b) Spesifikasi *Access Point*

Access Point yang digunakan pada penelitian ini yaitu salah satu perangkat UBNT, Pico Station M2HP dengan spesifikasi sebagai berikut (www.ubnt.com/airmax#picostation):

Processor : Atheros MIPS 24KC, 400 MHz

Memory : 32 MB SDRAM, 8 MB Flash

Networking Interface : (1) 10/100 Ethernet Port

Enclosure Characteristic : Outdoor UV Stabilized Plastic

Antenna Signal : (1) External, 5 dBi Omni Antenna

Operating Frequency : 2412-2462 MHz

Range	: Indoor Up to 200m Outdoor Up to 500m
Power Consumption	: 8W
Power Supply	: 15V, 0.8A Power Adapter
Output power	: 28dBm

2. Spesifikasi Software

Tabel 4.1. Spesifikasi Software

No	Software	Keterangan
1	Ubuntu 13.04	Sistem Operasi untuk <i>Server Streaming</i>
2	Win 7 Ultimate	Sistem Operasi untuk <i>Client</i>
3	Win 8 Ultimate	Sistem Operasi untuk <i>Client</i>
4	VLC Media Player 2.0.7	Pemutar Video <i>Client Server</i>
5	Wireshark 1.10	Tools Menangkap Paket <i>Video Streaming</i>

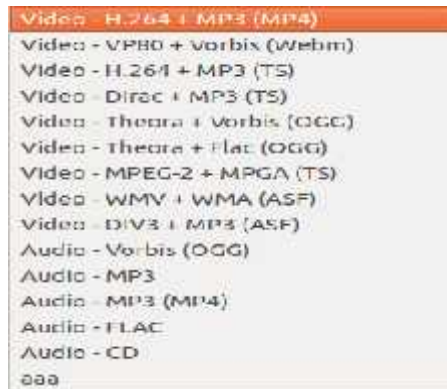
3. Web Server

Web Server merupakan sebuah perangkat lunak *Server* yang berfungsi menerima permintaan *Hypertext Transfer Protocol* (HTTP) dari *client* yang dikenal dengan mesin pencari (web browser) dan mengirimkan kembali hasilnya dalam bentuk halaman web yang berbentuk dokumen *Hypertext Markup Language* (HTML). Halaman web mempunyai alamat *Internet protocol* (IP) baik *private* maupun *public* yang sering disebut dengan website sehingga client dapat mengakses halaman web dengan mengakses alamat tersebut. *Web Server* dapat dibangun pada berbagai *platform* dengan aplikasinya masing-masing. Pada penelitian ini *web Server* berjalan diatas *platform* Linux Ubuntu 13.04 dengan *software* Apache *Web Server*, *web*

Server ini digunakan sebagai *Server* yang menyimpan beberapa file dengan kapasitas besar.

4. Codec Video Streaming

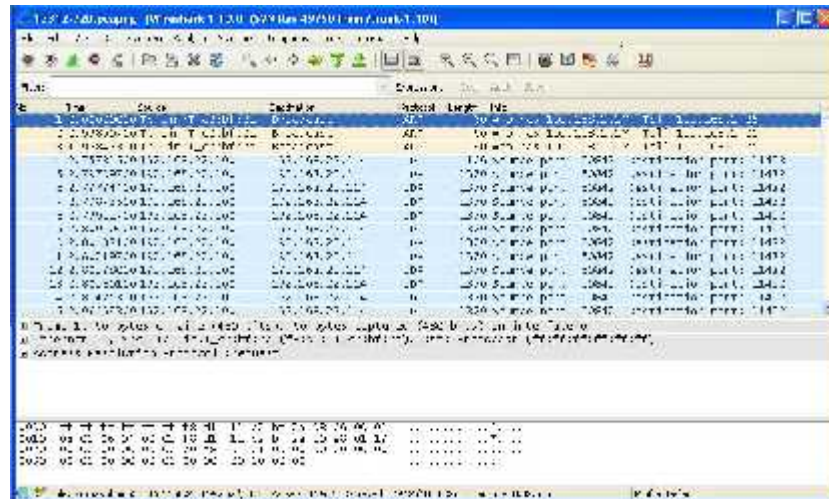
Codec merupakan suatu cara untuk merubah sinyal audio dan video menjadi bit digital yang kemudian akan ditransmisikan ke jaringan dalam bentuk paket IP. Pada penelitian ini digunakan codec VLC Media Player. VLC merupakan perangkat lunak *free* yang dikembangkan oleh VideoLAN Organization yang *support* beberapa protokol komunikasi untuk *video streaming*. Pada penelitian ini VLC berjalan di dua *platform*, Linux Ubuntu 13.04 dan Windows.



Gambar 4.1. Transcoding VLC Media Player

5. Paket *Sniffer* menggunakan Wireshark

Paket snifferi merupakan sebuah cara untuk menganalisa kondisi jaringan komputer. Pada penelitian ini digunakan Wireshark sebagai tools paket *sniffer*. Wireshark mampu menangkap paket data/informasi yang berjalan di jaringan yang akan dianalisa. Semua jenis paket informasi dalam berbagai format protokol pun akan dengan mudah ditangkap dan dianalisa.



Gambar 4.2. Interface Wireshark

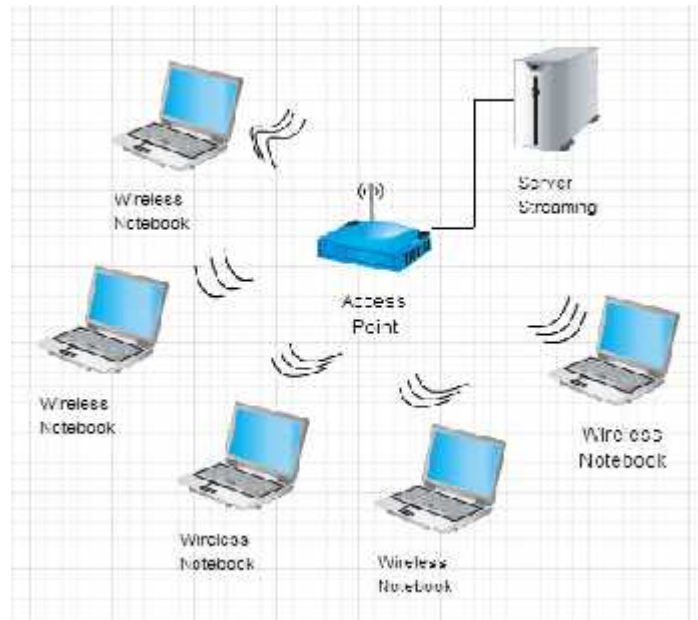
IV.2. PERANCANGAN

IV.2.1. Perancangan Infrastruktur

Sebelum membangun sebuah jaringan *wireless* ada beberapa langkah yang harus dilakukan sehingga jaringan yang dibuat dapat bekerja dengan baik dan data-data yang didapat sesuai dengan yang diharapkan. Penulis mengumpulkan seluruh perangkat yang dibutuhkan.

Jaringan *wireless* yang akan dirancang terdiri dari satu buah *Wireless* yang nantinya sebagai AP (Access Point), satu buah *Server streaming* dan lima buah *Client*. *Server* dan AP akan saling terhubung sehingga membentuk sebuah infrastruktur jaringan dimana *client* dapat terhubung dan melakukan akses konten ke *Server streaming*. Berikut adalah topologi jaringan yang akan dibangun untuk pengukuran *Quality of Service*.

IV.2.1.1. Topologi Jaringan



Gambar 4.3. Topologi Jaringan

Terlihat pada Gambar 4.3 merupakan topologi jaringan yang akan dirancang dimana *Server streaming* terhubung ke AP (Access Point) dengan mode *bridge* yang nantinya sebagai penghubung antara *client* dengan *Server straming*. Topologi ini yang nantinya digunakan untuk mengukur kinerja *Quality of Service* dari jaringan *wireless*.

IV.2.1.2. IP Address

Tabel 4.2. IP Address

Perangkat	Interface	IP Address	Gateway
<i>Server Streaming</i>	Local	192.168.22.100	-
AP	Local	192.168.22.101	192.168.22.100
<i>Client 1,2,3,4,5</i>	Ethernet	DHCP	-

Tabel 4.2. IP Address adalah alokasi IP Address yang akan dirancang nantinya, di sisi *Server* dan (AP) Access point konfigurasi IP Address secara *static*.

Terlihat ditabel 4.2 pada AP (*Access Point*) konfigurasi *Gateway* mengarah ke IP *Address Server* . Sedangkan pada sisi *client* pengaturan IP *Address* nya DHCP (otomatis didapat dari sisi *Server*).

IV.2.1.3. Skenario Pengujian

IV.2.1.3.1. Skenario I

Pengujian dilakukan tidak menggunakan manajemen *bandwidth*, *Server* menjalankan Video dengan resolusi video 720p, memiliki *bitrate* video 2413kbps dan *bitrate* audio 191kbps. Datarate yang diberikan untuk perbandingan yaitu 8192kbps, 9216kbps, 10240kbps dan 11264kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.2. Skenario II

Pengujian dilakukan tidak menggunakan manajemen *bandwidth*, *Server* menjalankan Video dengan resolusi video 480p, memiliki *bitrate* video 751kbps dan *bitrate* audio 128kbps. Datarate yang diberikan untuk perbandingan yaitu 4096kbps, 5120kbps, 6144kbps dan 6656kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.3. Skenario III

Pengujian dilakukan tidak menggunakan manajemen *bandwidth*, *Server* menjalankan Video dengan resolusi video 360p, memiliki *bitrate* video 593kbps dan *bitrate* audio 96kbps. Datarate yang diberikan untuk perbandingan yaitu 2048kbps, 2560kbps, 3072kbps dan 4096kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.4. Skenario IV

Pengujian dilakukan tidak menggunakan manajemen *bandwidth*, *Server* menjalankan Video dengan resolusi video 240p, memiliki *bitrate* video 498kbps dan *bitrate* audio 96kbps. Datarate yang diberikan untuk perbandingan yaitu 1024kbps, 1280kbps, 1535kbps dan 1792kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.5. Skenario V

Pengujian dilakukan menggunakan manajemen *bandwidth* dan dibagi menjadi 2 kelas, kelas 1(semua traffic video menuju port 11432) dan kelas 2 (semua traffic http menuju port 80) Antar kelas dapat memakai *bandwidth* yang sedang tidak digunakan. *Server* menjalankan Video dengan resolusi video 720p, memiliki *bitrate* video 2413kbps dan *bitrate* audio 191kbps. Datarate yang diberikan untuk perbandingan yaitu 8192kbps, 9216kbps, 10240kbps dan 11264kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.6. Skenario VI

Pengujian dilakukan menggunakan manajemen *bandwidth* dan dibagi menjadi 2 kelas, kelas 1(semua traffic video menuju port 11432) dan kelas 2 (semua traffic http menuju port 80) Antar kelas dapat memakai *bandwidth* yang sedang tidak digunakan. *Server* menjalankan Video dengan resolusi video 480p, memiliki *bitrate* video 751kbps dan *bitrate* audio 128kbps. Datarate yang diberikan untuk perbandingan yaitu 4096kbps, 5120kbps, 6144kbps dan 6656kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.7. Skenario VII

Pengujian dilakukan menggunakan manajemen *bandwidth* dan dibagi menjadi 2 kelas, kelas 1(semua traffic video menuju port 11432) dan kelas 2 (semua traffic http menuju port 80) Antar kelas dapat memakai *bandwidth* yang sedang tidak digunakan. *Server* menjalankan Video dengan resolusi video 360p, memiliki *bitrate* video 593kbps dan *bitrate* audio 96kbps. Datarate yang diberikan untuk perbandingan yaitu 2048kbps, 2560kbps, 3072kbps dan 4096kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

IV.2.1.3.8. Skenario VIII

Pengujian dilakukan menggunakan manajemen *bandwidth* dan dibagi menjadi 2 kelas, kelas 1(semua traffic video menuju port 11432) dan kelas 2 (semua traffic http menuju port 80) Antar kelas dapat memakai *bandwidth* yang sedang tidak digunakan. *Server* menjalankan Video dengan resolusi video 240p, memiliki *bitrate* video 498kbps dan *bitrate* audio 96kbps. Datarate yang diberikan untuk perbandingan yaitu 1024kbps, 1280kbps, 1535kbps dan 1792kbps. Diakses oleh 5 *client* menggunakan *notebook* secara bersamaan. Akan dilakukan Pengukuran *Delay*, *Throughput*, *Packet Loss* dan *Jitter*.

BAB V

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini, penulis akan mengimplementasikan hasil rancangan yang telah dibuat. Peneliti akan melakukan beberapa pengujian parameter dari *Quality of Service* terhadap beban data *Video Streaming* dan melakukan evaluasi terhadap hasil yang didapat. Parameter yang akan di analisa yaitu *Delay*, *Jitter*, *Lacket loss*, dan *Troughput* dari suatu jaringan dengan 4 tipe resolusi video yaitu 720p, 480p, 360p dan 240p. Pengujian dilakukan untuk mendapatkan *bandwidth* minimal dari keempat tipe resolusi video. Peneliti menganalisa hasil data yang didapat yang nantinya akan menghasilkan beberapa kesimpulan dari penelitian tersebut.

V.1. IMPLEMENTASI

V.1.1. Konfigurasi Jaringan *Wireless*

Menghubungkan komponen jaringan sesuai dengan topologi yang telah dirancang. Server *Streaming* dihubungkan ke AP (*Access Point*) melalui kabel LAN yang telah dibuat dengan mode kabel *straight* sehingga terjalin koneksi antar bagian tersebut. Kemudian melakukan konfigurasi *IP Address* pada Server *Streaming* dan AP (*Access Point*) yang telah dijelaskan pada Tabel 4.2. *IP Address*. Untuk memastikan koneksi sudah berjalan baik perlu dilakukan test koneksi berupa *Ping* antar bagian yang terhubung.

V.1.2. Konfigurasi Pico Station M2HP

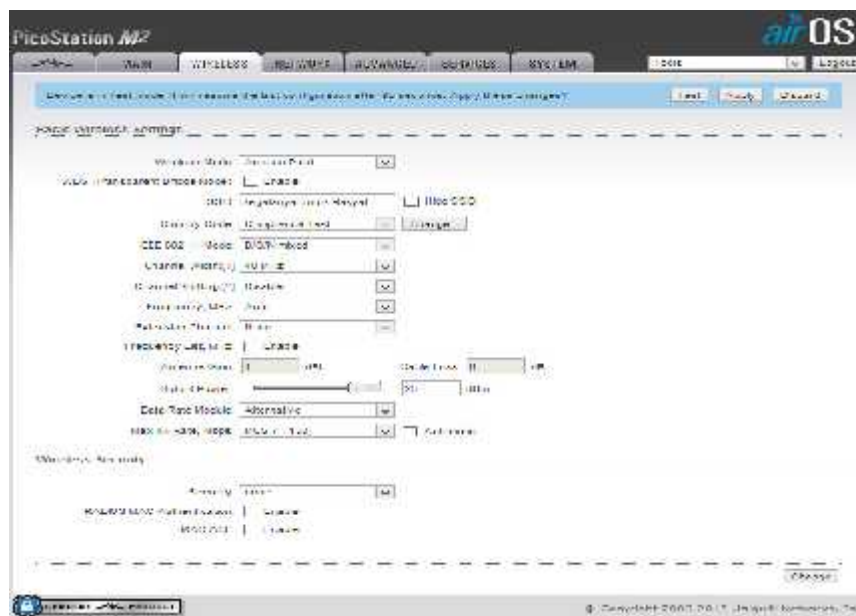
Setelah infrastruktur jaringan *wireless* dibangun dan saling terhubung satu dengan lain nya maka tahap selanjutnya dilakukan konfigurasi terhadap *Acces Point*, berikut langkah – langkah nya :

1. Login Perangkat dengan mengakses IP Address perangkat. Akan muncul tampilan beranda dari Pico Station. Masukkan *Username* dan *Password* (*username: ubnt* *password: ubnt*).



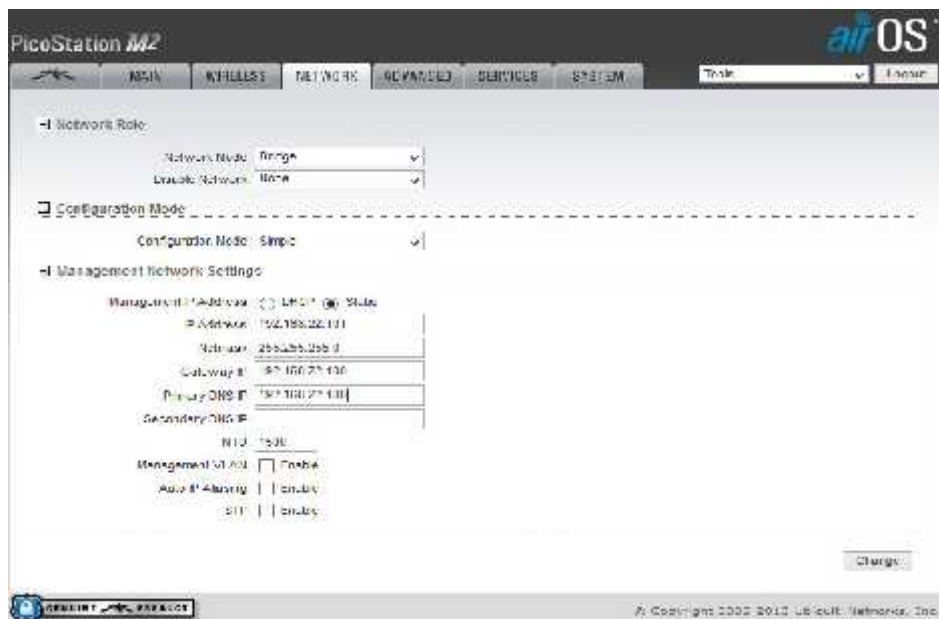
Gambar 5.1. Tampilan *web login Access Point*

2. Setelah *login* masuk ke tab *Wireless*. Lalu setting seperti dibawah ini , lalu klik tombol *Change*.



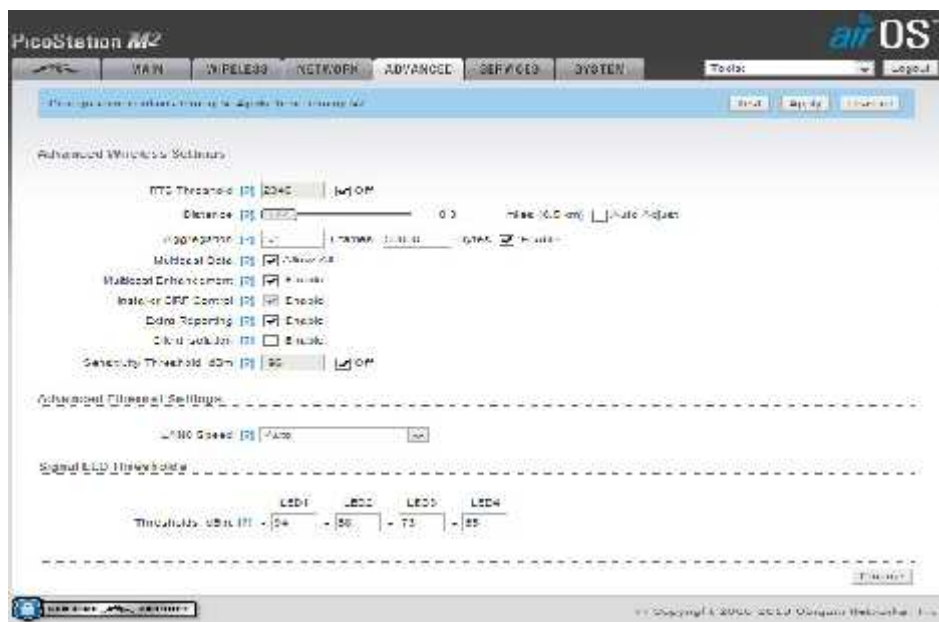
Gambar 5.2. Tampilan konfigurasi *Wireless*

3. Setelah itu masuk ke *tab Network*, rubah pengaturan pada *network* seperti dibawah ini : Lalu klik tombol *Change*.



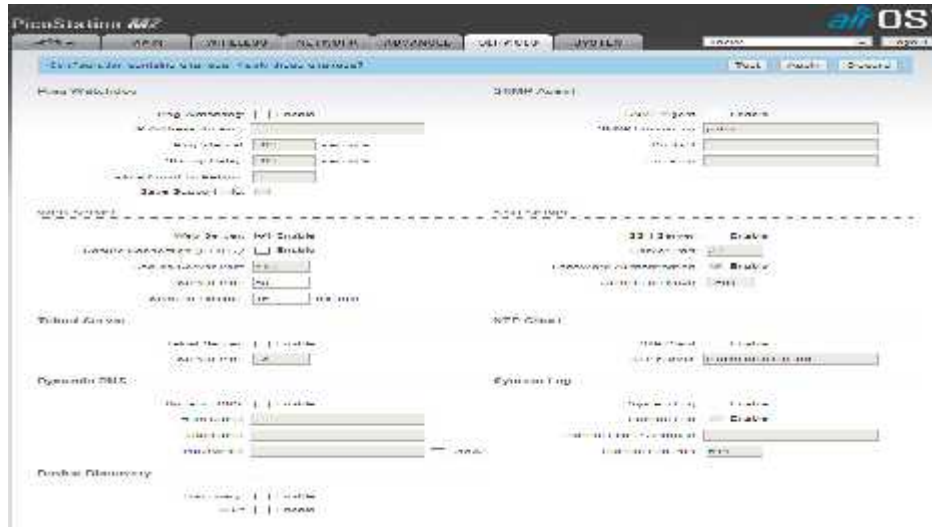
Gambar 5.3. Tampilan konfigurasi *Network*

4. Lanjut ke *tab Advance*, *set distance* untuk jangkauan *Access point*. Dan ceklist beberapa pengaturan seperti gambar dibawah ini. Setelah itu klik tombol *Change*.



Gambar 5.4. Tampilan konfigurasi *Advance*

5. Setelah itu masuk ke tab *Service*. Ini merupakan *service* pada perangkat *access point*. Pada tab ini cukup menceklist *webserver* saja. Sisa nya biarkan saja tidak terceklist. Lalu klik tombol *Change*.



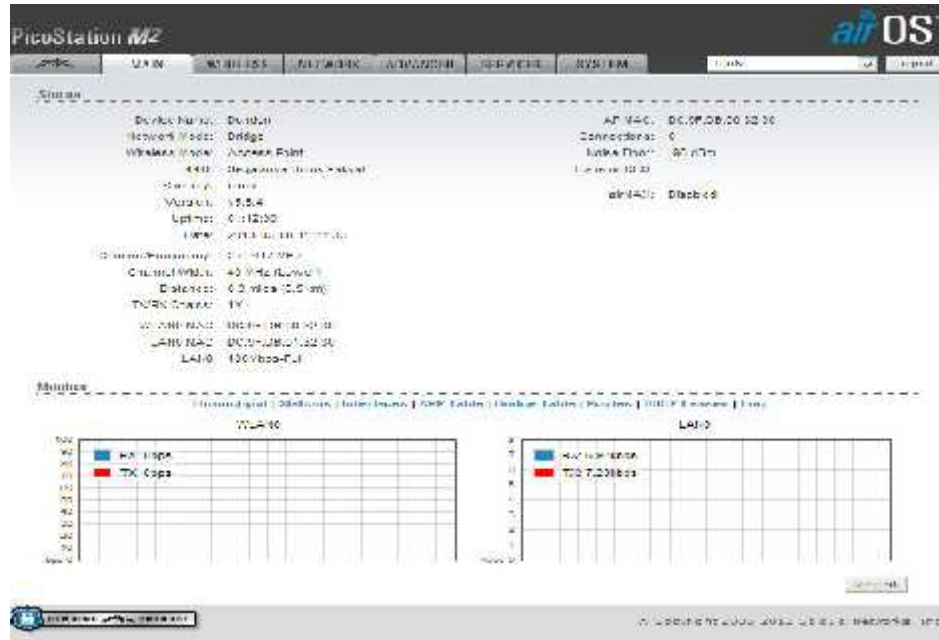
Gambar 5.5. Tampilan konfigurasi *Services*

6. Lanjut ke tab terakhir yaitu *System*. Pada tab ini yang ubah yaitu *Device Name* (*Optional*) dan *system account*. Ganti *username* dan *password* nya. Setelah itu klik tombol *Change* dan klik tombol *Apply* berada di bawah bar *tab*.



Gambar 5.6. Tampilan konfigurasi *System*

7. Access Point akan melakukan restart beberapa detik. Tampilan *browser* akan blank sementara. Lalu akan menampilkan *tab* Main yang berisi status dari AP. Sampai tahap ini telah selesai konfigurasi *Access Point*.



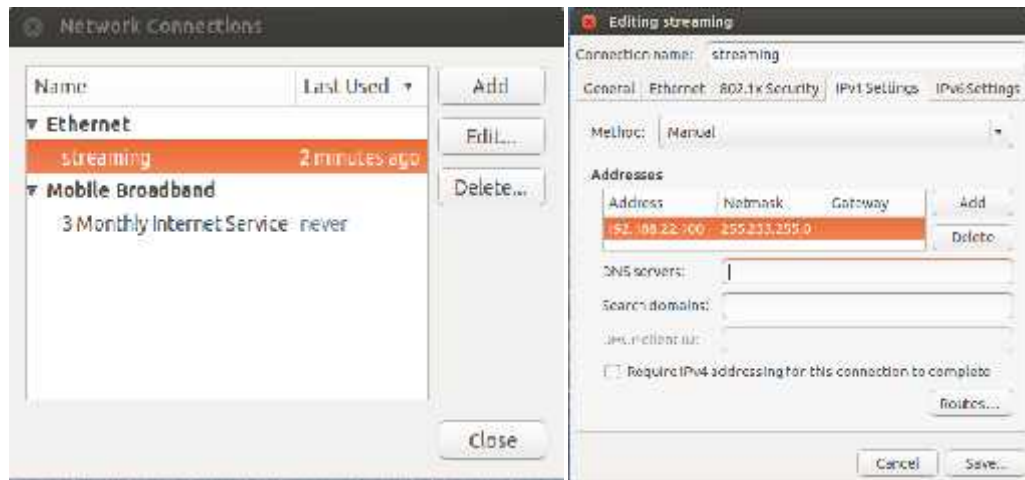
Gambar 5.7. Tampilan Status *Access Point*

V.1.3. Konfigurasi Server *Streaming*

Server *streaming* ini sebagai tempat sharing video dan sebagai manajemen *bandwidth* untuk *Client*. Adapun langkah – langkah nya yaitu :

Tahap ini yaitu proses penginstalan OS untuk server *Streaming*. OS yang digunakan peneliti yaitu Ubuntu 13.04. tahap selanjutnya yaitu dilakukan konfigurasi pada server, adapun langkah-langkah nya sebagai berikut :

1. Setting IP Address. Klik icon *Connection* di taskbar pojok atas lalu pilih *Edit Connection*. Akan muncul *window* baru dan klik tombol *add* lalu isi ip *address* sesuai dengan gambar dibawah ini.



Gambar 5.8. Tampilan konfigurasi IP Address

2. Update Sistem melalui terminal.

```
elroy@video-server:~$ apt-get update && apt-get upgrade
```

3. Install Komponen Web Server, DHCP Server, DNS Server dan Streaming Server

```
elroy@video-server:~$ apt-get install apache2 php5 vlc player dhcp3-server  
bind9
```

4. Menulis file baru sebagai Laman Web di /var/www/

```
elroy@video-server:~$ vim /var/www/index.html
```

5. Konfigurasi dhcp Server.

```
elroy@video-server:~# vim /etc/dhcp/dhcpd.conf
```

```
default lease time 600;  
max-lease-time 7200;  
  
option subnet-mask 255.255.255.0;  
option broadcast-address 192.168.22.255;  
option routers 192.168.22.100;  
option domain-name-servers 192.168.22.100;  
option domain-name "videocser.ver";  
  
subnet 192.168.22.0 netmask 255.255.255.0 {  
  range 192.168.22.101 192.168.22.105;  
}
```

Gambar 5.9. Konfigurasi DHCP

6. Konfigurasi DNS Server.

- a. Edit file named.conf.local.

```
elroy@video-server:~# vim /etc/bind/named.conf.local
```

```

// If you have multiple interfaces, you can add them here
//

// Consider adding the 1918 names here, if they are not listed in your
// organization
//include "/etc/bind/names.1918.conf";
zone "videoser.ver" {
    type master;
    file "/etc/bind/db.videoser";
};

zone "192.168.22.100" {
    type master;
    file "/etc/bind/db.192";
};

```

Gambar 5.10. Konfigurasi file named.conf.local

- b. Edit file db.videoser.

```
elroy@video-server:~# vim /etc/bind/db.videoser
```

```

// Data file for local machine
$TTL 604800
IN SOA videoser.ver. root.videoser.ver. (
    201401010001 1 10000 3600 604800 10000 )
IN NS videoser.ver.
IN A 192.168.22.100
www IN A 192.168.22.100
ftp IN A 192.168.22.100
video IN A 192.168.22.100
vamp IN A 192.168.22.100

```

Gambar 5.11. Konfigurasi file db.videoser

- c. Edit file db.192.

```
elroy@video-server:~# vim /etc/bind/db.192
```

```

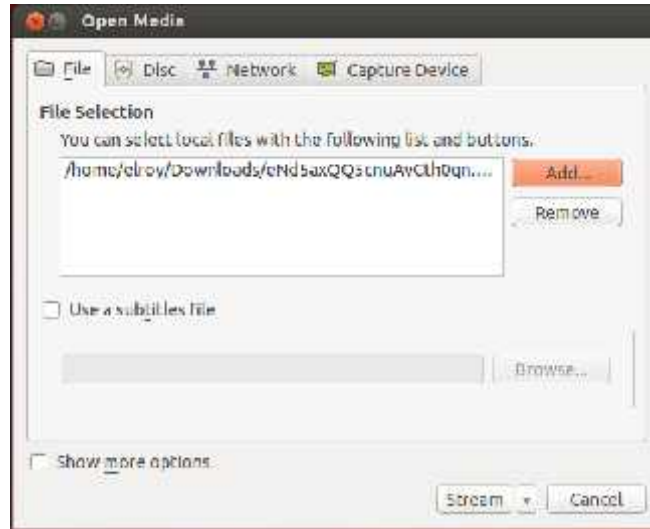
// Data file for local machine
$TTL 604800
IN SOA videoser.ver. root.videoser.ver. (
    201401010001 1 10000 3600 604800 10000 )
IN NS videoser.ver.
192.168.22.100 IN A 192.168.22.100

```

Gambar 5.12. Konfigurasi file db.192

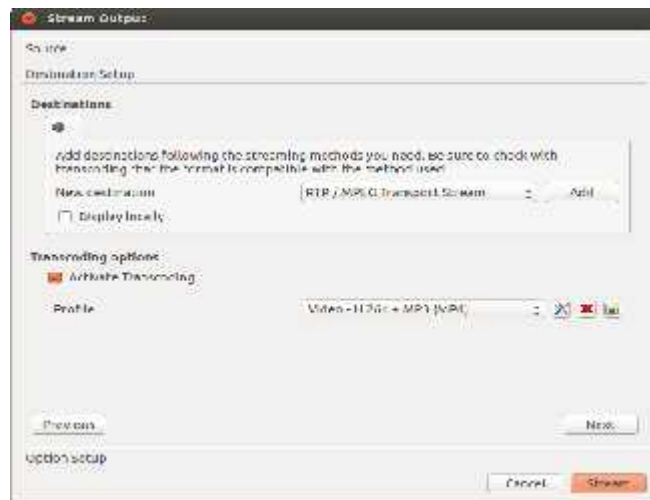
7. Setting VLC sebagai *Server Streaming*

- a. Buka vlc player.
- b. Pilih menu media pada menu bar lalu klik menu *stream*. Akan muncul tampilan seperti dibawah ini.



Gambar 5.13. Open Media Files VLC

- c. Klik add untuk memilih file yang akan di *stream*. Lalu klik *stream*.
- d. Klik next lalu pilih tipe yang akan di *stream*, pilih RTP. Lalu *add*.



Gambar 5.14. Konfigurasi protokol *Streaming*

- Stream output**

Source:

Destination:

☒ **RTSP**

This module outputs the transcoded stream on a network via RTSP.

Address:

Base port:

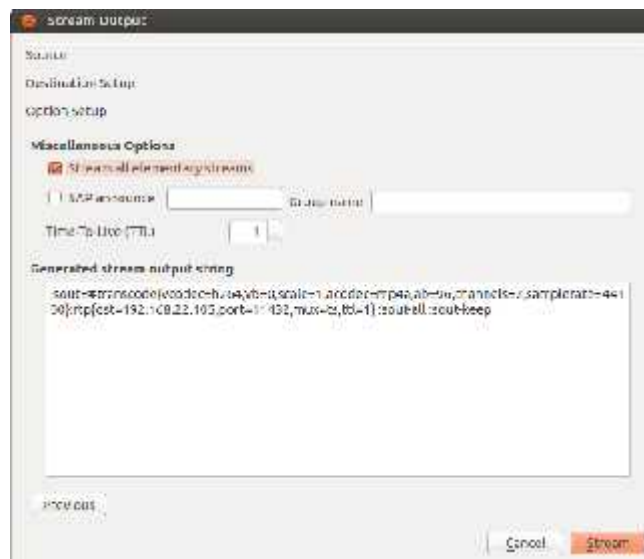
Transcoding options

☒ **All video transcoding**

Profile:

Option setup

f. Langkah terakhir ceklist pada *Stream all elementary streams*. Kemudian klik *Stream*.



8. Konfigurasi script tc menggunakan metode HTB
 - a. Buat file konfigurasi tc di direktori root dengan nama tc.sh

V-9

- b. Isi konfigurasi seperti dibawah ini lalu simpan dengan perintah :wq

```
# Name of the traffic control command.
TC=/sbin/tc
# The network interface we're planning on limiting bandwidth.
IF=eth0          # Interface
# Download limit (in mega bits)
DNLD=5120kbit    # DOWNLOAD Limit
# Upload limit (in mega bits)
UPLD=320kbit     # UPLOAD Limit
# IP address of the machine we are controlling
IP5=192.168.22.114
IP4=192.168.22.113
IP3=192.168.22.112
IP2=192.168.22.111
IP1=192.168.22.110
# Filter options for limiting the intended interface.
U322="$TC filter add dev $IF protocol ip parent 1:0 prio 2
u32"
U321="$TC filter add dev $IF protocol ip parent 1:0 prio 1
u32"
start() {
# We'll use Hierarchical Token Bucket (HTB) to shape
bandwidth.
$TC qdisc add dev $IF root handle 1: htb default 30
$TC class add dev $IF parent 1: classid 1:1 htb rate $DNLD

$TC class add dev $IF parent 1:1 classid 1:10 htb rate
1024kbit ceil 5120kbit
$TC class add dev $IF parent 1:10 classid 1:101 htb rate
128kbit ceil 256kbit
$TC class add dev $IF parent 1:10 classid 1:102 htb rate
896kbit ceil 1024kbit

$TC class add dev $IF parent 1:1 classid 1:11 htb rate
1024kbit ceil 5120kbit
```

```

$TC class add dev $IF parent 1:11 classid 1:111 htb rate
128kbit ceil 256kbit
$TC class add dev $IF parent 1:11 classid 1:112 htb rate
896kbit ceil 1024kbit
$TC class add dev $IF parent 1:1 classid 1:12 htb rate
1024kbit ceil 5120kbit
$TC class add dev $IF parent 1:12 classid 1:121 htb rate
128kbit ceil 256kbit
$TC class add dev $IF parent 1:12 classid 1:122 htb rate
896kbit ceil 1024kbit

$TC class add dev $IF parent 1:1 classid 1:13 htb rate
1024kbit ceil 5120kbit
$TC class add dev $IF parent 1:13 classid 1:131 htb rate
128kbit ceil 256kbit
$TC class add dev $IF parent 1:13 classid 1:132 htb rate
896kbit ceil 1024kbit

$TC class add dev $IF parent 1:1 classid 1:14 htb rate
1024kbit ceil 5120kbit
$TC class add dev $IF parent 1:14 classid 1:141 htb rate
128kbit ceil 256kbit
$TC class add dev $IF parent 1:14 classid 1:142 htb rate
896kbit ceil 1024kbit

$TC class add dev $IF parent 1: classid 1:2 htb rate $UPLD
$TC class add dev $IF parent 1:2 classid 1:21 htb rate 64kbit
ceil 320kbit
$TC class add dev $IF parent 1:2 classid 1:22 htb rate 64kbit
ceil 320kbit
$TC class add dev $IF parent 1:2 classid 1:23 htb rate 64kbit
ceil 320kbit
$TC class add dev $IF parent 1:2 classid 1:24 htb rate 64kbit
ceil 320kbit

```

```

$TC class add dev $IF parent 1:2 classid 1:25 htb rate 64kbit
ceil 320kbit

$U321 match ip src $IP2/32 flowid 1:21
$U321 match ip src $IP3/32 flowid 1:22
$U321 match ip src $IP4/32 flowid 1:23
$U321 match ip src $IP5/32 flowid 1:24
$U321 match ip src $IP6/32 flowid 1:25

$U322 match ip dst $IP2/32 match ip sport 80 0xffff flowid
1:101
$U321 match ip dst $IP2/32 match ip dport 11432 0xffff flowid
1:102

$U322 match ip dst $IP3/32 match ip sport 80 0xffff flowid
1:111
$U321 match ip dst $IP3/32 match ip dport 11432 0xffff flowid
1:112

$U322 match ip dst $IP4/32 match ip sport 80 0xffff flowid
1:121
$U321 match ip dst $IP4/32 match ip dport 11432 0xffff flowid
1:122

$U322 match ip dst $IP5/32 match ip sport 80 0xffff flowid
1:131
$U321 match ip dst $IP5/32 match ip dport 11432 0xffff flowid
1:132

$U322 match ip dst $IP6/32 match ip sport 80 0xffff flowid
1:141
$U321 match ip dst $IP6/32 match ip dport 11432 0xffff flowid
1:142

}

```

```

stop() {
# Stop the bandwidth shaping.
    $TC qdisc del dev $IF root
}
restart() {
# Self-explanatory.
    stop
    sleep 1
    start
}
show() {

# Display status of traffic control status.
    $TC -s qdisc ls dev $IF
}
case "$1" in
    start)
        echo -n "Starting bandwidth shaping: "
        start
        echo "done"
        ;;
    stop)
        echo -n "Stopping bandwidth shaping: "
        stop
        echo "done"
        ;;
    restart)
        echo -n "Restarting bandwidth shaping: "
        restart
        echo "done"
        ;;
    show)
        echo "Bandwidth shaping status for $IF:"
        show
        echo " "

```

```

    ;;
    * )
        pwd=$(pwd)
    echo "Usage: tc.bash {start|stop|restart|show}"
    ;;

esac

exit 0

```

- c. Ubah hak akses file tc.sh jadi *executable*.

```
elroy@video-server:~# chmod +x ./tc.sh
```

- d. Jalankan tc.sh

```
elroy@video-server:~# ./tc.sh start
```

V.1.4. Konfigurasi *Client*

Tahap ini adalah tahap terakhir dari konfigurasi, yang perlu di konfigurasi pada sisi *Client* yaitu :

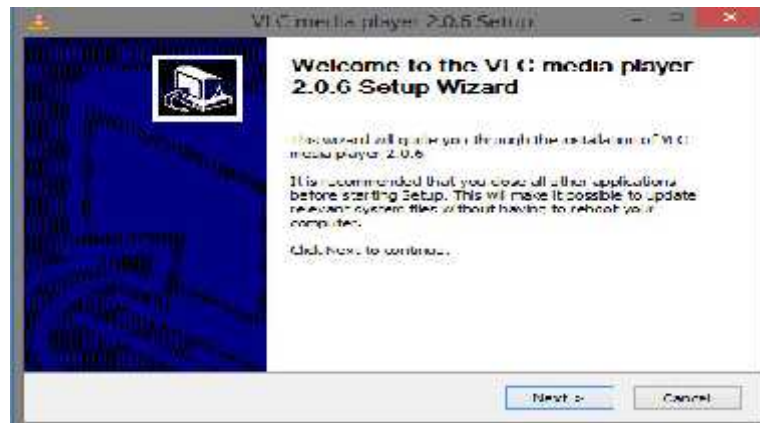
- a. Install VLC Media Player

1. *Download* VLC Media Player versi terbaru di *homepage* nya, yaitu <http://www.videolan.org/vlc/download-windows.html>.
2. Setelah download selesai, klik file vlc-2.0.6-win64.exe lalu jalankan proses installasinya.



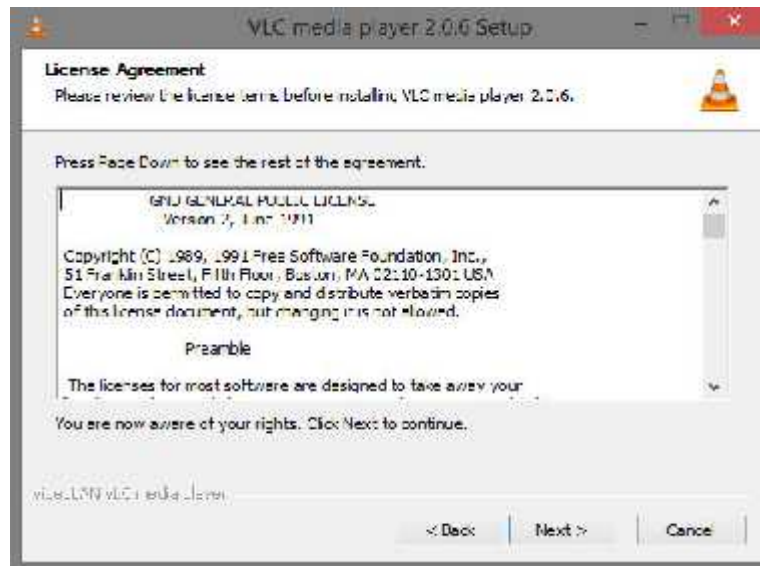
Gambar 5.17. VLC Installer

3. Pada saat muncul tampilan *wizard* instalasi VLC media player klik *next* untuk melanjutkan.



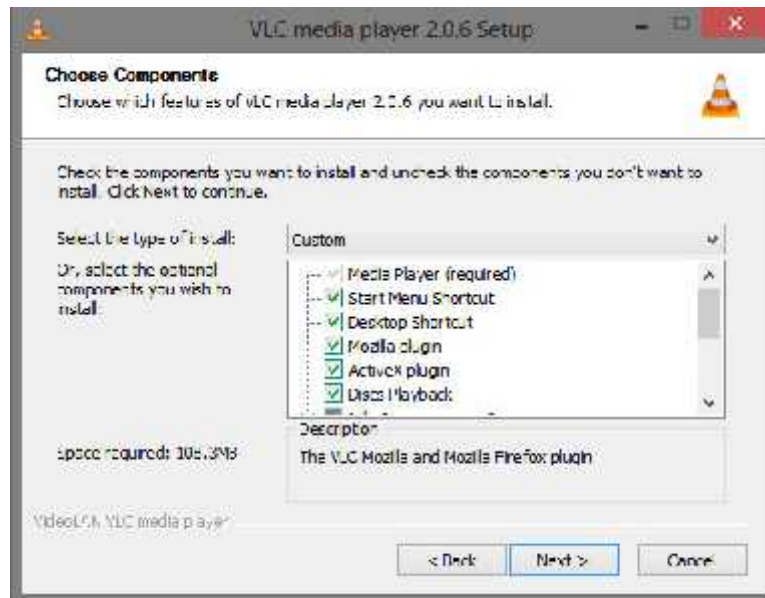
Gambar 5.18. *Installation Wizard* VLC Media Player

4. Akan ditujukan ke *License Agreement*, yaitu Persetujuan instalasi. Klik *next* untuk melanjutkan.



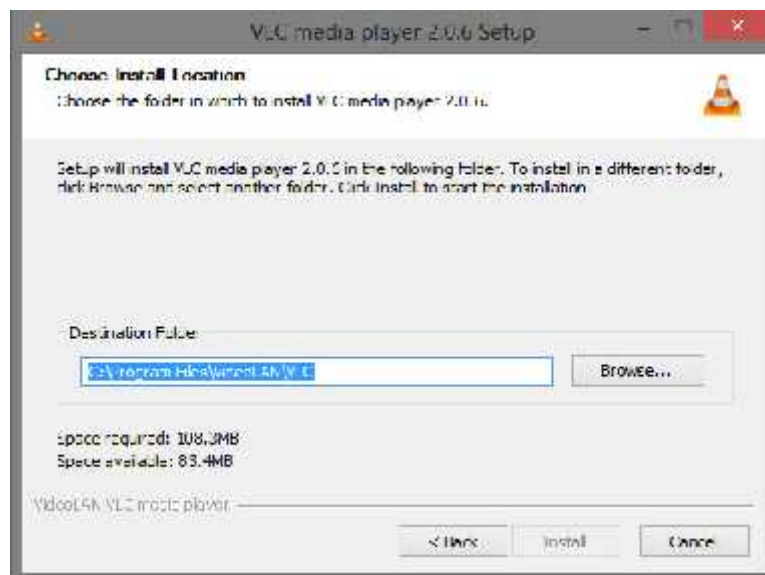
Gambar 5.19. *User Agreement* VLC Media player

5. Berikut adalah komponen yang akan di install, klik *next* untuk melanjutkan.



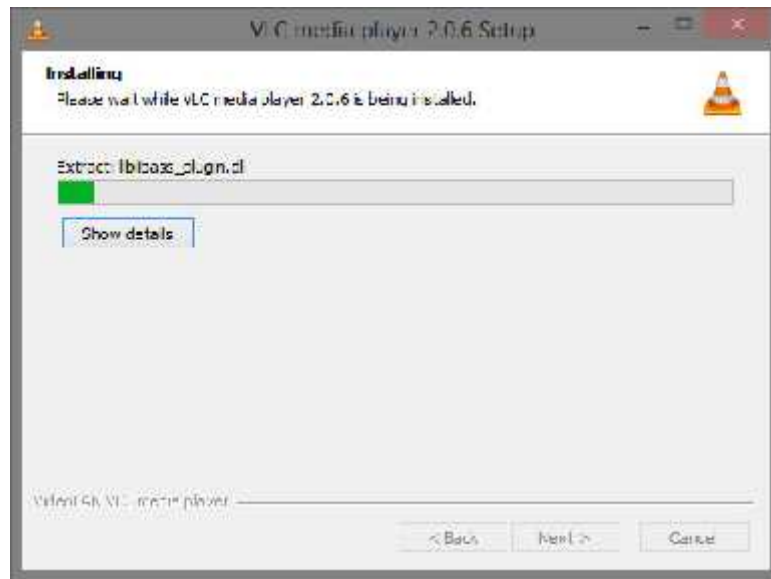
Gambar 5.20. *Component Installation* VLC Media player

6. Pilih lokasi dimana VLC media player akan di *install*. Lalu klik *install*.



Gambar 5.21. *Destination Installation* VLC Media Player

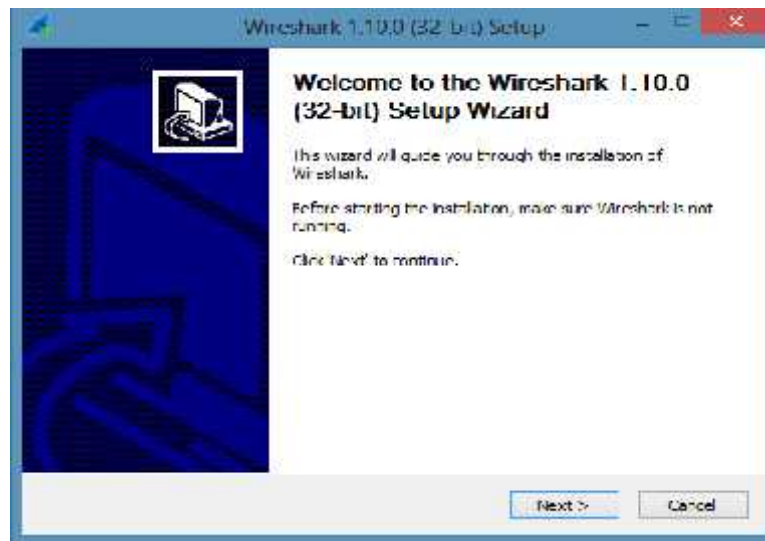
7. Tunggu beberapa saat proses instalasi sedang berjalan. Setelah selesai klik *finish*.



Gambar 4.22. *Progress Installation VLC Media Player*

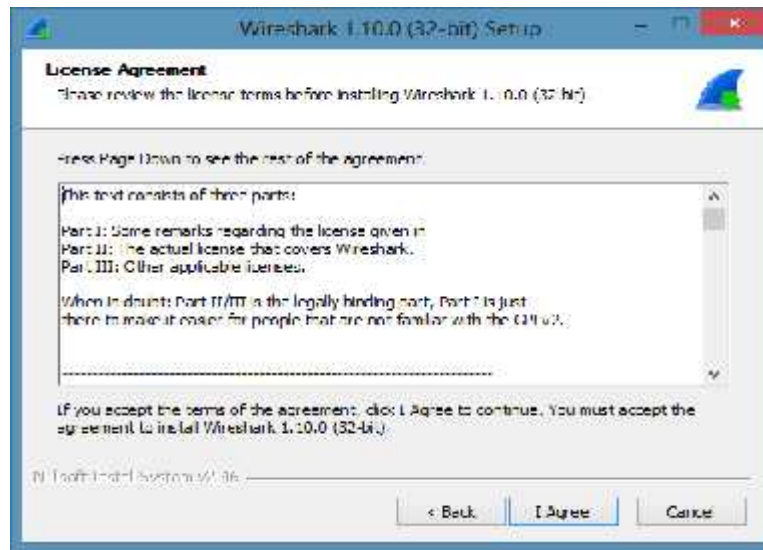
b. Install Wireshark

1. *Download* wireshark versi terupdate di *homepage* nya yaitu <http://www.wireshark.org/download.html>
2. Setelah selesai *download* klik 2x file setup wireshark. Akan muncul tampilan *wizard* seperti dibawah ini.



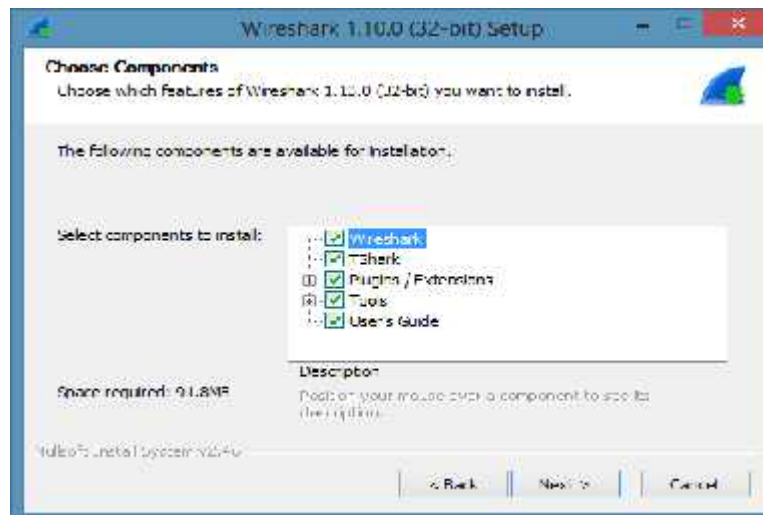
Gambar 5.23. *Wizard Installation Wireshark*

3. Klik next lalu muncul tampilan *User Agreement*. klik *I Agree*.



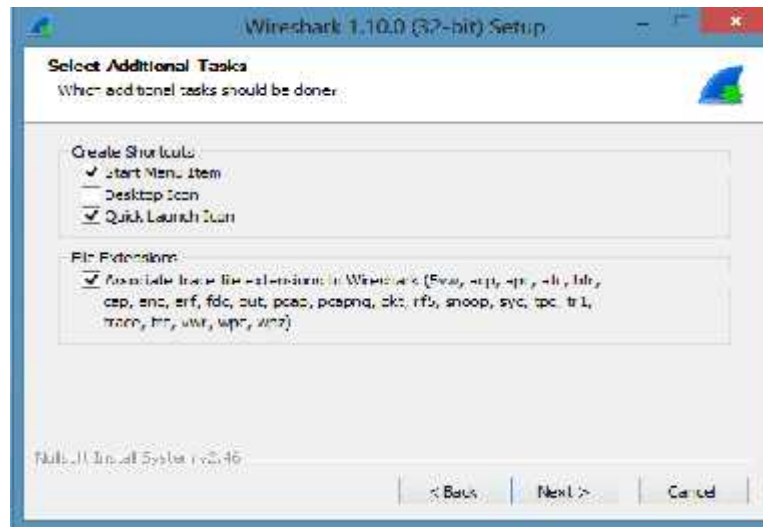
Gambar 4.24. *User Aggrement Installation Wireshark*

4. Ceklist semua komponen yang ada lalu klik *next* untuk melanjutkan.



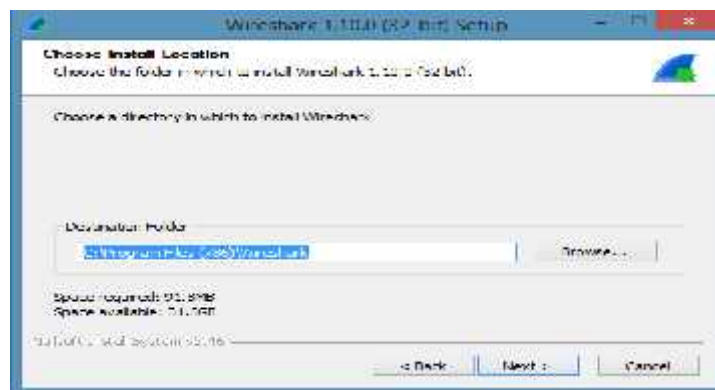
Gambar 5.25. *Wizard Component Installation Wireshark*

5. Ceklist *file extensions* dan ceklist start menu item untuk membuat *shortcut* di *start menu*. Kemudian klik *next*.



Gambar 5.26. Wizard Additional Tasks Wireshark

6. Pilih tempat penyimpanan file yang akan di *install*. Kemudian klik *next*.



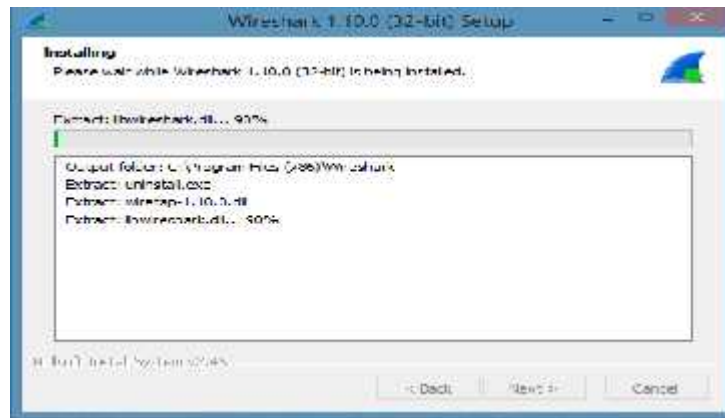
Gambar 5.27. Destination Folder Wireshark

7. Ceklist WinPcap jika belum terinstall. Kemudian klik *install*



Gambar 5.28. Install WinPcap

8. Proses instalasi sedang berjalan, tunggu beberapa saat. Jika selesai klik tombol *finish*.



Gambar 5.29. *Progress Installation Wireshark*

V.1.5. Action Taking (Melakukan Tindakan)

Selanjutnya setelah model dibuat berdasarkan rancangan, maka akan dilakukan tahap implementasi yaitu menjalankan *Streaming* video pada sisi server dengan VLC v.2.0.7, *Source* video yang di siarkan ada 4 tipe yaitu dimulai dari Resolusi 240p, 360p, 480p dan 720p. *Transcoding* yang digunakan adalah *video codec* H.264. Video dijalankan menggunakan protokol RTP (*Real-Time Protocol*) *multicast* sehingga pada sisi *Client* mengakses video secara *realtime*. *Client* mengakses Video *Streaming* dengan membuka *Web Server* yang telah di konfigurasi pada *server*. Akan dilakukan beberapa analisa dari komponen dari jaringan yaitu *Delay*, *Jitter*, *Packet loss*, dan *Troughput* ketika dibebani oleh *streaming* video dengan kualitas video tertentu. Akan dilakukan juga perbandingan menggunakan HTB dan tanpa HTB. Pengujian dilakukan menggunakan *tools wireshark* pada sisi *client* untuk menangkap paket pada saat video dijalankan.

V.1.5.1. Langkah Pengambilan Data

a. Sisi Server

1. Menjalankan *service* DHCP

```
elroy@video-server:~# /etc/init.d/isc-dhcp-server start
```

2. Menjalankan *Service* DNS

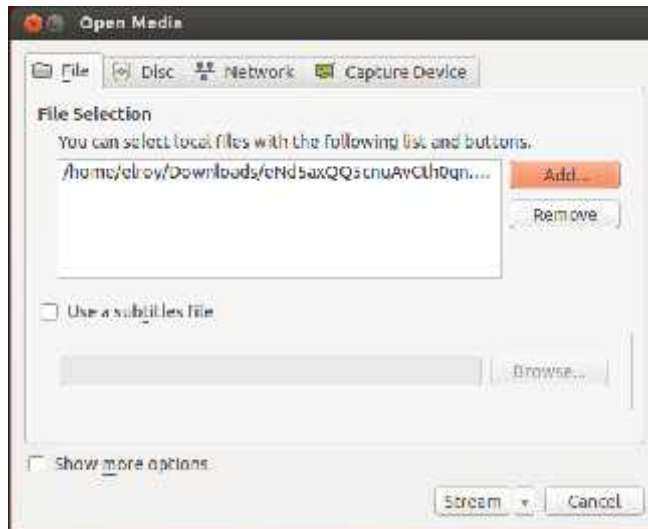
```
elroy@video-server:~# /etc/init.d/bind9 start
```

3. Menjalankan *Service* WEB

```
elroy@video-server:~# /etc/init.d/apache2 start
```

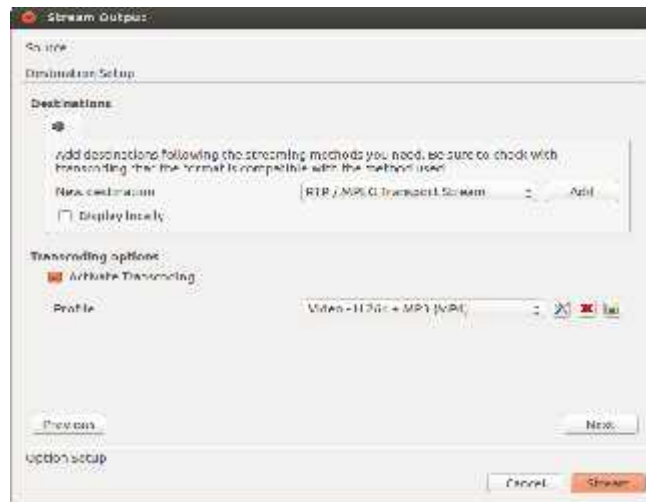
4. Menjalankan VLC Media Player

5. Memilih file video yang akan di *streaming*



Gambar 5.30. memilih file video yang akan di *stream*

6. Memilih protokol RTP/MPEG *Transport Stream* dan *transcoding* Video-H.264 + MP3 (MP4).



Gambar 5.31. protokol dan *transcoding video streaming*

7. Memasukkan IP Address berapa saja yang dapat melakukan *streaming* beserta *port* nya.



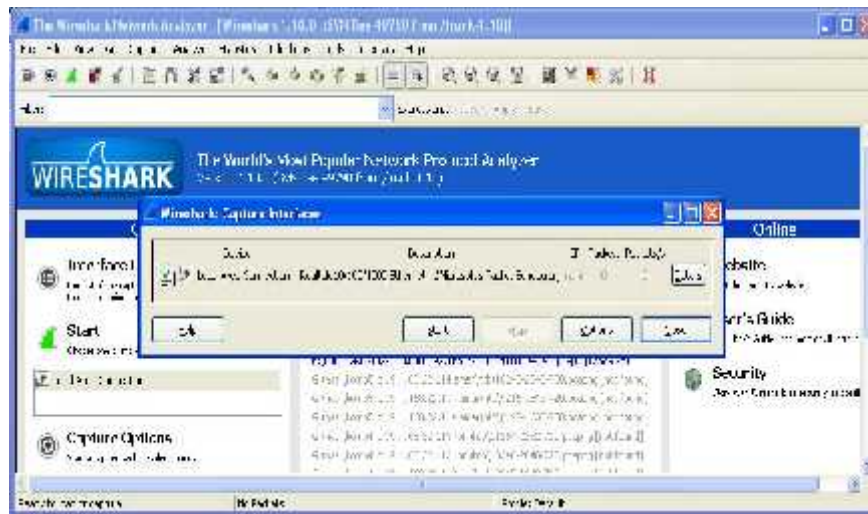
Gambar 5.32. IP Address dan *port* yang digunakan

8. Ceklist *Stream all elementary streams* lalu lakukan *stream*.
9. Setelah 1menit 30detik klik tombol *stop*.

b. Sisi *Client*

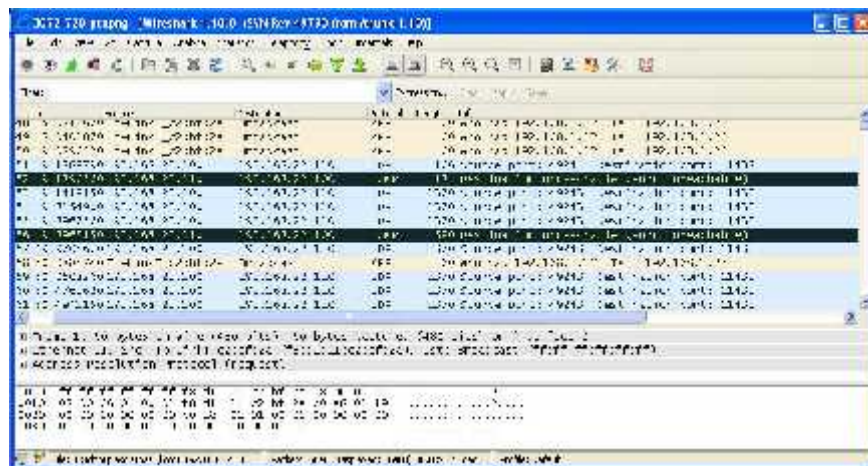
1. Sambungkan koneksi *Wireless client* ke AP (*Access Point*.)
2. Buka aplikasi *Wireshark*.

3. Jalankan proses *sniffing* dengan menceklist *device Wireless* lalu klik start



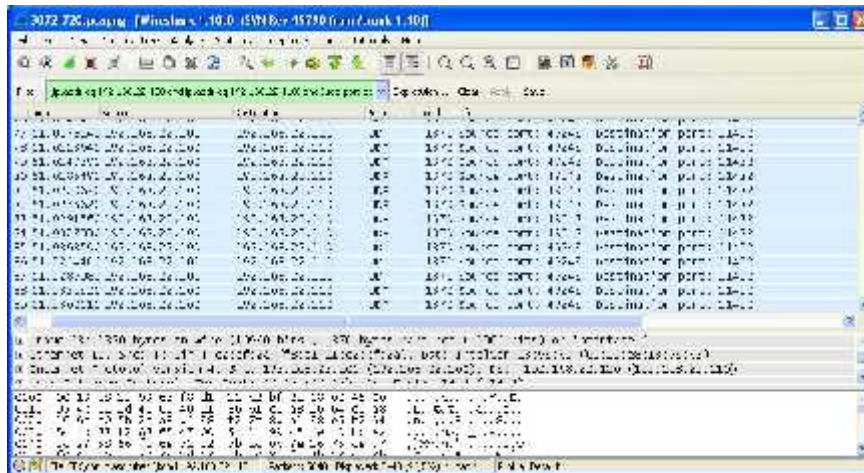
Gambar 5.33. Proses Sniffing

4. Buka alamat videoser.ver/wp untuk melakukan *streaming*.
5. Jika video sudah berhenti klik stop untuk berhenti melakukan *sniffing* jaringan.



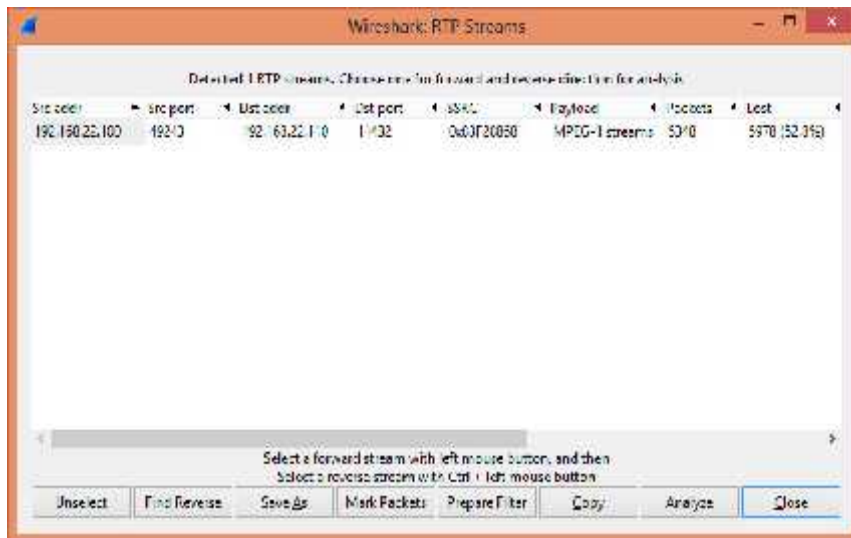
Gambar 5.34. Hasil Sniffing paket video streaming

6. Filter paket UDP dengan klik kanan pada salah satu paket UDP lalu pilih follow UDP stream



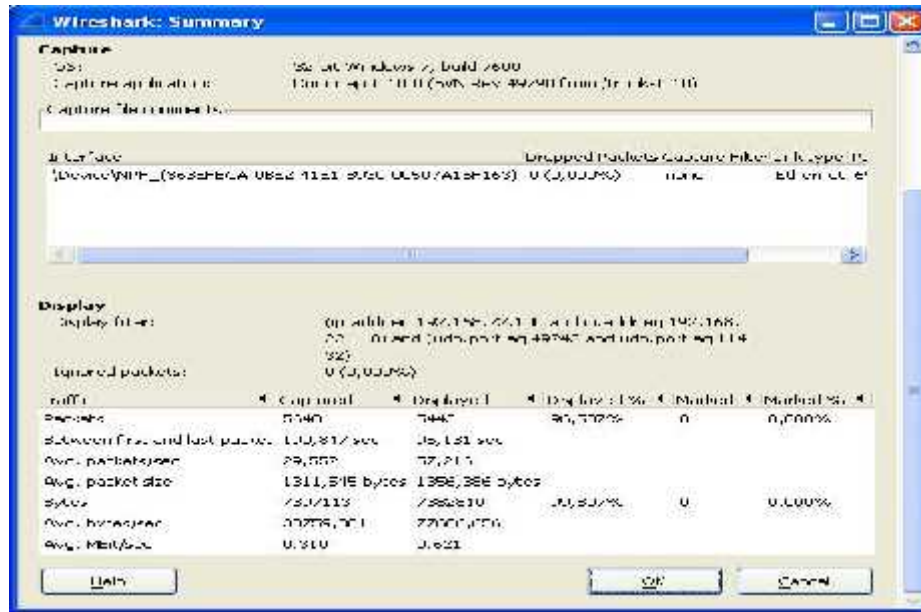
Gambar 5.35. Hasil *Filter UDP*

7. *Decode* paket UDP ke RTP dengan mengklik kanan salah satu paket udp lalu pilih *decode as* dan pilih RTP.
8. Untuk melihat nilai *Packet Loss* pilih menu *Telephony* lalu pilih RTP kemudian pilih *show all streams*



Gambar 5.36. nilai *packet loss* yang didapat

9. Buka statistik untuk mendapatkan data hasil *sniffing*.



Gambar 5.37. Statistik hasil *sniffing*

10. Menghitung *Packet loss*

Menghitung *paket loss* dengan rumus :

$$\text{Packet Loss} = \frac{(\text{paket dikirim} - \text{paket diterima})}{\text{Paket dikirim}} \times 100\%$$

$$\begin{aligned} \text{Packet Loss} &= \frac{11326 - 5348}{11326} \times 100\% \\ &= 52,8\% \end{aligned}$$

11. Menghitung *Delay*

Menghitung nilai *delay* dengan rumus :

$$\text{Delay rata-rata} = \frac{\text{Total Delay}}{\text{Total Paket yang diterima}}$$

$$\begin{aligned} \text{Delay rata-rata} &= \frac{95,131}{5348} \\ &= 0,0177881451009723 \text{ s} \\ &= 18 \text{ ms} \end{aligned}$$

12. Menghitung *Jitter*

Menghitung nilai *Jitter* dengan rumus :

$$Jitter = \frac{(\text{Total variasi } delay)}{\text{Total paket yang diterima} - 1}$$

$$\text{Variasi } delay = (\text{delay } n - \text{delay } (n-1))$$

$$\begin{aligned} Jitter &= \frac{-0.670051}{5442} \\ &= 0.000130566 \text{ s} \\ &= 0.130566 \text{ ms} \end{aligned}$$

13. Menghitung *Throughput*

Menghitung nilai *Throughput* dengan rumus :

$$Throughput = \frac{\text{jumlah data yang dikirim}}{\text{waktu pengiriman data}}$$

$$\begin{aligned} &= \frac{7382810}{95,131} \\ &= 77606,77 \text{ bytes/s} \\ &= 620854,16 \text{ bit/s} \\ &= 0,62085 \text{ Mbit/s} \end{aligned}$$

V.2. PENGUJIAN

Tabel 5.1. Pengujian skenario I dengan *bandwidth* 8192kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	8.1%	8.77 ms	0 ms	1.243kbps
2	8.4%	8.8 ms	0 ms	1.232kbps
3	8.4%	8.82 ms	0 ms	1.231kbps
4	9.7%	9.02 ms	0 ms	1.215kbps
5	10%	9.11 ms	0 ms	1.203kbps

Tabel 5.2. Pengujian skenario I dengan *bandwidth* 9216kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	8.2%	8.6 ms	0 ms	1.317kbps
2	8.5%	8.7 ms	0 ms	1.259kbps
3	8.8%	8.75 ms	0 ms	1.252kbps
4	8.9%	8.76 ms	0 ms	1.251kbps
5	9.8%	8.85 ms	0 ms	1.238kbps

Tabel 5.3. Pengujian skenario I dengan *bandwidth* 10240kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	5.2%	8.4 ms	0 ms	1.287kbps
2	6.0%	8.6 ms	0 ms	1.272kbps
3	6.1%	8.64 ms	0 ms	1.268kbps
4	6.7%	8.69 ms	0 ms	1.260kbps
5	6.7%	8.70 ms	0 ms	1.258kbps

Tabel 5.4. Pengujian skenario I dengan *bandwidth* 11264kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	4.9%	8.30 ms	0 ms	1.307kbps
2	5.6%	8.51 ms	0 ms	1.287kbps
3	5.6%	8.52 ms	0 ms	1.286kbps
4	6.0%	8.56 ms	0 ms	1.280kbps
5	6.7%	8.6 ms	0 ms	1.269kbps

Tabel 5.5. Pengujian skenario II dengan *bandwidth* 4096kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	14%	15.01 ms	0 ms	724kbps
2	14.2%	15.11 ms	0 ms	717kbps
3	14.6%	15.34 ms	0 ms	714kbps
4	15.2%	15.46 ms	0 ms	709kbps
5	16.1%	15.57 ms	0 ms	695kbps

Tabel 5.6. Pengujian skenario II dengan *bandwidth* 5120kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	6.8%	13.7 ms	0 ms	792kbps
2	7.1%	13.95 ms	0 ms	785kbps
3	7.2%	13.97 ms	0 ms	784kbps
4	7.5%	14.02 ms	0 ms	782kbps
5	8.2%	14.12 ms	0 ms	776kbps

Tabel 5.7. Pengujian skenario II dengan *bandwidth* 6144kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	3.6%	13.23	0 ms	828kbps
2	3.4%	13.30	0 ms	824kbps
3	3.5%	13.33 ms	0 ms	822kbps
4	4.0%	13.39	0 ms	818kbps
5	4.0%	13.40 ms	0 ms	818kbps

Tabel 5.8. Pengujian skenario II dengan *bandwidth* 6656kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	2.5%	13.26 ms	0 ms	826kbps
2	2.9%	13.31 ms	0 ms	823kbps
3	3.1%	13.35 ms	0 ms	821kbps
4	3.4%	13.38 ms	0 ms	819kbps
5	3.6%	13.41 ms	0 ms	817kbps

Tabel 5.9. Pengujian skenario III dengan *bandwidth* 2048kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	28.6%	27.29 ms	0 ms	402kbps
2	28.4%	27.25 ms	0 ms	402kbps
3	28.9%	27.43 ms	0 ms	400kbps
4	29.2%	27.54 ms	0 ms	398kbps
5	29.6%	27.69 ms	0 ms	396kbps

Tabel 5.10. Pengujian skenario III dengan *bandwidth* 2560kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	15.5%	22.26 ms	0 ms	492kbps
2	15.7%	22.28 ms	0 ms	492kbps
3	16.1%	22.40 ms	0 ms	489kbps
4	16.3%	22.45 ms	0 ms	488kbps
5	16.3%	22.48 ms	0 ms	487kbps

Tabel 5.11. Pengujian skenario III dengan *bandwidth* 3072kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	7.6%	20.37 ms	0 ms	538kbps
2	7.7%	20.40 ms	0 ms	537kbps
3	8.4%	20.53 ms	0 ms	534kbps
4	8.2%	20.48 ms	0 ms	535kbps
5	8.5%	20.57 ms	0 ms	533kbps

Tabel 5.12. Pengujian skenario III dengan *bandwidth* 4096kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	1.4%	18.64 ms	0 ms	588kbps
2	1.5%	18.66 ms	0 ms	587kbps
3	1.7%	18.71 ms	0 ms	586kbps
4	1.6%	18.69 ms	0 ms	586kbps
5	2.2%	18.79 ms	0 ms	583kbps

Tabel 5.13. Pengujian skenario IV dengan *bandwidth* 1024kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	33.5%	53.22 ms	0 ms	206kbps
2	33.9%	53.38 ms	0 ms	205kbps
3	34.2%	53.52 ms	0 ms	205kbps
4	34.0%	53.64 ms	0 ms	204kbps
5	34.1%	53.67 ms	0 ms	204kbps

Tabel 5.14. Pengujian skenario IV dengan *bandwidth* 1280kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	19.1%	42.88 ms	0 ms	256kbps
2	19.2%	42.90 ms	0 ms	255kbps
3	19.7%	43.12 ms	0 ms	254kbps
4	19.3%	42.93 ms	0 ms	255kbps
5	19.8%	43.11 ms	0 ms	254kbps

Tabel 5.15. Pengujian skenario IV dengan *bandwidth* 1535kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	5.9%	36.85 ms	0 ms	297kbps
2	5.7%	36.80 ms	0 ms	298kbps
3	6.1%	36.94 ms	0 ms	297kbps
4	5.8%	36.83 ms	0 ms	298kbps
5	6.3%	36.95 ms	0 ms	297kbps

Tabel 5.16. Pengujian skenario IV dengan *bandwidth* 1792kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	0%	33.30 ms	0 ms	329kbps
2	0%	33.31 ms	0 ms	329kbps
3	0.2%	33.36 ms	0 ms	328kbps
4	0%	33.31 ms	0 ms	329kbps
5	0.5%	33.46 ms	0 ms	328kbps

Tabel 5.17. Pengujian skenario V dengan *bandwidth* 8192kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	9.9%	8.97 ms	0 ms	1.222kbps
2	10%	8.97 ms	0 ms	1.221kbps
3	10%	8.97 ms	0 ms	1.221kbps
4	10%	8.97 ms	0 ms	1.221kbps
5	10%	8.97 ms	0 ms	1.221kbps

Tabel 5.18. Pengujian skenario V dengan *bandwidth* 9216kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	5.4%	8.4 ms	0 ms	1.297kbps
2	5.4%	8.4 ms	0 ms	1.297kbps
3	5.4%	8.4 ms	0 ms	1.296kbps
4	5.4%	8.4 ms	0 ms	1.297kbps
5	5.4%	8.4 ms	0 ms	1.296kbps

Tabel 5.19. Pengujian skenario V dengan *bandwidth* 10240kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	5%	8.4 ms	0 ms	1.291kbps
2	3.7%	8.37 ms	0 ms	1.309kbps
3	3.7%	8.37 ms	0 ms	1.309kbps
4	3.7%	8.37 ms	0 ms	1.309kbps
5	3.7%	8.37 ms	0 ms	1.309kbps

Tabel 5.20. Pengujian skenario V dengan *bandwidth* 11264kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	2.8%	8.21 ms	0 ms	1.335kbps
2	2.8%	8.21 ms	0 ms	1.335kbps
3	2.8%	8.21 ms	0 ms	1.335kbps
4	2.8%	8.21 ms	0 ms	1.335kbps
5	2.8%	8.21 ms	0 ms	1.335kbps

Tabel 5.21. Pengujian skenario VI dengan *bandwidth* 4096kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	14.6%	15.13 ms	0 ms	724kbps
2	14.6%	15.24 ms	0 ms	719kbps
3	14.6%	15.24 ms	0 ms	719kbps
4	14.6%	15.24 ms	0 ms	719kbps
5	14.6%	15.24 ms	0 ms	719kbps

Tabel 5.22. Pengujian skenario VI dengan *bandwidth* 5120kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	7.3%	13.98 ms	0 ms	784kbps
2	7.3%	13.98 ms	0 ms	784kbps
3	7.3%	13.98 ms	0 ms	784kbps
4	7.3%	13.98 ms	0 ms	784kbps
5	7.3%	13.98 ms	0 ms	784kbps

Tabel 5.23. Pengujian skenario VI dengan *bandwidth* 6144kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	3.6%	13.23 ms	0 ms	828kbps
2	3.6%	13.23 ms	0 ms	828kbps
3	3.6%	13.23 ms	0 ms	828kbps
4	3.6%	13.23 ms	0 ms	828kbps
5	3.6%	13.23 ms	0 ms	828kbps

Tabel 5.24. Pengujian skenario VI dengan *bandwidth* 6656kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	3%	13.30 ms	0 ms	823kbps
2	2.9%	13.30 ms	0 ms	824kbps
3	2.9%	13.30 ms	0 ms	824kbps
4	2.9%	13.30 ms	0 ms	824kbps
5	2.9%	13.30 ms	0 ms	824kbps

Tabel 5.25. Pengujian skenario VII dengan *bandwidth* 2048kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	28.6%	27.43 ms	0 ms	399kbps
2	28.6%	27.43 ms	0 ms	399kbps
3	28.6%	27.43 ms	0 ms	399kbps
4	28.6%	27.43 ms	0 ms	399kbps
5	28.6%	27.43 ms	0 ms	399kbps

Tabel 5.26. Pengujian skenario VII dengan *bandwidth* 2560kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	15.7%	22.33 ms	0 ms	491kbps
2	15.7%	22.33 ms	0 ms	491kbps
3	15.7%	22.33 ms	0 ms	491kbps
4	15.7%	22.33 ms	0 ms	491kbps
5	15.7%	22.33 ms	0 ms	491kbps

Tabel 5.27. Pengujian skenario VII dengan *bandwidth* 3072kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	8%	20.46 ms	0 ms	536kbps
2	8%	20.46 ms	0 ms	536kbps
3	8%	20.46 ms	0 ms	536kbps
4	8%	20.46 ms	0 ms	536kbps
5	8%	20.46 ms	0 ms	536kbps

Tabel 5.28. Pengujian skenario VII dengan *bandwidth* 4096kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	1.5%	18.92 ms	0 ms	579kbps
2	1.5%	18.92 ms	0 ms	579kbps
3	1.5%	18.92 ms	0 ms	579kbps
4	1.5%	18.92 ms	0 ms	579kbps
5	1.5%	18.92 ms	0 ms	579kbps

Tabel 5.29. Pengujian skenario VIII dengan *bandwidth* 1024kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	33.5%	53.64 ms	0 ms	204kbps
2	33.5%	53.64 ms	0 ms	204kbps
3	33.5%	53.64 ms	0 ms	204kbps
4	33.5%	53.64 ms	0 ms	204kbps
5	33.5%	53.64 ms	0 ms	204kbps

Tabel 5.30. Pengujian skenario VIII dengan *bandwidth* 1280kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	18.6%	43.71 ms	0 ms	251kbps
2	18.6%	43.71 ms	0 ms	251kbps
3	18.6%	43.71 ms	0 ms	251kbps
4	18.6%	43.71 ms	0 ms	251kbps
5	18.6%	43.71 ms	0 ms	251kbps

Tabel 5.31. Pengujian skenario VIII dengan *bandwidth* 1535kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	5.8%	36.84 ms	0 ms	297kbps
2	5.8%	36.84 ms	0 ms	297kbps
3	5.8%	36.84 ms	0 ms	297kbps
4	5.8%	36.84 ms	0 ms	297kbps
5	5.8%	36.84 ms	0 ms	297kbps

Tabel 5.32. Pengujian skenario VIII dengan *bandwidth* 1792kbps

<i>Client</i>	<i>Packet Loss</i>	<i>Delay</i>	<i>Jitter</i>	<i>Throughput</i>
1	0.1%	32.71 ms	0 ms	335kbps
2	0.1%	32.71 ms	0 ms	335kbps
3	0.1%	32.71 ms	0 ms	335kbps
4	0.1%	32.71 ms	0 ms	335kbps
5	0.1%	32.71 ms	0 ms	335kbps

V.3. Evaluating (Melakukan Evaluasi)

Video dengan resolusi 720p agar dapat berjalan dengan baik pada *client*, membutuhkan *bandwidth* 11264kbps ketika tidak menerapkan HTB *client* tidak mendapatkan *bandwidth* untuk data teks. jika menerapkan HTB, agar dapat berjalan dengan baik pada *client* membutuhkan *bandwidth* 10240kbps dan *client* mendapatkan *bandwidth* untuk data teks. Video dengan resolusi 480p agar dapat berjalan dengan baik pada *client*, membutuhkan *bandwidth* 6656kbps ketika tidak menerapkan HTB *client* tidak mendapatkan *bandwidth* untuk data teks, jika menerapkan HTB, agar dapat berjalan dengan baik pada *client* membutuhkan *bandwidth* 6144kbps dan *client* mendapatkan *bandwidth* untuk data teks. Video dengan resolusi 360p agar dapat berjalan dengan baik pada *client*, membutuhkan *bandwidth* 4096kbps ketika tidak

menerapkan HTB *client* tidak mendapatkan *bandwidth* untuk data teks, jika menerapkan HTB, agar dapat berjalan dengan baik pada *client* membutuhkan *bandwidth* 4096kbps dan *client* mendapatkan *bandwidth* untuk data teks. Video dengan resolusi 240p agar dapat berjalan dengan baik pada *client*, membutuhkan *bandwidth* 1792kbps ketika tidak menerapkan HTB, jika menerapkan HTB *client* tidak mendapatkan *bandwidth* untuk data teks, agar dapat berjalan dengan baik pada *client* membutuhkan *bandwidth* 1792kbps dan *client* mendapatkan *bandwidth* untuk data teks.

Analisa hasil dari pengujian ini menunjukkan bahwa toleransi maksimal *packet loss* berdasarkan kualitas gambar dan suara yang di dapat berada pada nilai 3%. Jika lebih dari 3%, kualitas gambar dan suara yang didapat pada saat streaming video tidak memuaskan karena terlalu banyak *flicker* pada saat video berjalan. Hal ini dikarenakan protokol RTP tidak akan melakukan *retransmission* jika ada data yang hilang seperti protokol TCP yang melakukan pengecekan pada *frame header packet* jika ada paket yang hilang akan di request kembali.

Delay yang terjadi pada saat video dijalankan menurut peneliti hanya signifikan dimata pengguna. *Delay* ini bersifat relatif. Seorang pengguna tidak akan mempermasalahkan *delay* jika video yang di jalankan itu tidak membutuhkan waktu real-time untuk ditonton. Berbeda jika video yang dijalankan adalah *video conference* dan *videophone* yang signifikan nilai *delay* terhadap kualitas *video* dan *audio* yang ditonton atau didengar. Berdasarkan acuan TIPHON delay dibawah 150ms tergolong kategori bagus.

Pada realita nya *throughput* sangat berpengaruh, namun dalam hal ini jumlah *client* yang terkoneksi masing-masing mendapatkan *bandwidth* yang optimal. Nilai *throughput* akan terasa signifikan jika *throughput* yang didapat lebih kecil daripada *total bitrate* video yang akan menyebabkan nilai *packet loss* naik.

nilai *Jitter* di semua pengujian 0 ms, hal ini dikarenakan *streaming* video menggunakan protokol RTP (*Real-Time Transfer Protocol*) yang berjalan diatas UDP karena TCP tidak dapat beradaptasi pada pengiriman data yang *real-time*. RTP didesain untuk mengkompensasi *Jitter* dan *desequencing* yang terjadi pada jaringan. Frame header RTP memiliki dua buah komponen yang mengatur tiap individu dari *endpoint* ke *endpoint* yaitu *timestamp* dan *sequence number* yang berfungsi untuk mengatur suara percakapan agar terdengar seperti sebagaimana diucapkan dan untuk pengurutan paket data, mendeteksi adanya paket yang hilang. Jika paket RTP hilang (*lost*) atau di drop di jaringan maka RTP tidak akan melakukan *retransmission* agar tidak terjadi *delay* dikarenakan permintaan *retransmission*.

Dari hasil pengujian yang dilakukan perbandingan *streaming* video menggunakan HTB dan tanpa HTB. *Bandwidth* yang didapat tiap *client* merata, sehingga *Packet loss* dan *Delay* tiap *client* nilainya lebih kecil dan stabil dibandingkan dengan tanpa HTB. Hal ini dikarenakan tiap *client* sudah diberikan jatah *bandwidth* sehingga pada tiap tidak saling berebut *bandwidth* di jaringan. Dengan dilakukan manajemen *bandwidth* tiap *client* jika ada *client* yang melakukan *streaming* dan browsing ataupun download tidak akan mengganggu pengguna yang sedang *streaming* video, hal ini dikarenakan tiap *client* sudah diberikan *bandwidth* dan *bandwidth* yang diberikan pada tiap *client* tersebut dibagi lagi menjadi dua kelas, untuk *streaming* video dan untuk *browsing*.

Banyak faktor yang mempengaruhi kualitas dari video, pada kasus ini secara global dibagi 2, yaitu faktor jaringan dan faktor dari *Hardware*. Baik *Hardware* AP (*Access Point*) maupun *Hardware* dari sisi *client*. Konfigurasi *software* perlu juga diperhatikan, contohnya *client* dengan processor corei7 dan ram 4gb ddr3 video akan tetap terasa tersendat jika banyak program yang berjalan dan menghabiskan *resource* CPU. *Encoding* juga mempengaruhi kualitas dari video. Pastikan video yang diputar dengan *encoding* tersebut bisa berjalan dengan lancar dan hasil gambar tidak ada *flicker*.

V.1. Learning (Pembelajaran)

Tahap ini merupakan bagian akhir siklus yang telah dilalui dengan melaksanakan review tahap-pertahap yang telah berakhir kemudian penelitian ini dapat berakhir. Beberapa attribut dari Qos yang telah didapat nilai nya akan di review dan akan didapat suatu kesimpulan yang akan menjadi standar untuk penelitian ini. Dan hasil penelitian ini juga dapat dipertimbangkan untuk tindakan kedepan yang bisa dilakukan dalam kaitan dengan penelitian ini.

BAB VI

KESIMPULAN DAN SARAN

VI.1. KESIMPULAN

1. Untuk mendapatkan hasil yang bagus pada video, *packet loss* harus berkisar antara 0% - 3%. Jika *packet loss* lebih dari 3% hasil video yang didapatkan tidak memuaskan karena video mengalami banyak kerusakan pada gambar dan suara.
2. Nilai *Jitter* di semua pengujian 0 ms, hal ini dikarenakan *streaming* video menggunakan protokol RTP (*Real-Time Transfer Protocol*) yang berjalan diatas UDP. Protokol UDP ini sifatnya hanya mengirim paket-paket tersebut tanpa melakukan pengecekan apakah paket yang dikirim sampai pada tujuan atau hilang. RTP didesain untuk mengkompensasi *jitter* dan *desequencing* yang terjadi pada jaringan. Jika diukur variasi *delay* antara pengiriman paket pertama dan kedua nilainya dibawah 0 ms.
3. Untuk *streaming* video yang tidak terlalu membutuhkan resolusi yang detail sebaiknya gunakan *transcoding*/kompresi video seminim mungkin agar penggunaan *Bandwidth* lebih sedikit.
4. Dari hasil penelitian ini, protokol RTP tidak direkomendasikan sebagai media *streaming* melalui *internet*. Melihat jaringan *internet* sangat luas dan terdapat banyak *hops* sehingga diragukan video akan sampai pada pengguna dengan kualitas yang bagus.
5. Faktor-faktor yang bisa mempengaruhi QoS pada Jaringan adalah redaman, *distorsi* dan *noise*. Kapasitas *Bandwidth* yang tersedia juga berpengaruh terhadap QoS.

6. *Quality of Service* (QoS) bukan membatasi tetapi lebih kepada menjaga kualitas *Bandwidth*, tanpa adanya *Quality of Service* dalam sebuah Jaringan mengakibatkan ketidaksinambungan *Bandwidth* yang diterima *client*.
7. Hierarchical Token Bucket (HTB) merupakan teknik QoS yang mampu memaksimalkan *Bandwidth* yang tidak terpakai, sehingga kualitas pelayanan menjadi lebih meningkat, Hasil yang dicapai yaitu Setiap paket memiliki *Bandwidth* minimal. Setiap paket dapat memperoleh *Bandwidth* lebih dari *Bandwidth* minimal tetapi tidak melebihi *Bandwidth* maksimal, Selama *traffic* pada parentnya tidak penuh. Terjadi pemerataan *Bandwidth* sesuai prioritasnya saat kondisi *traffic* seluruh paket penuh.

VI.2. SARAN

Untuk pengembangan penelitian ini disarankan :

1. Melakukan pengujian menggunakan *smartphone* dalam mengakses *video streaming* dari *server*. Melihat hasil parameter Qos pada *smartphone*.
2. Mencari cara agar *web live streaming* yang dibangun dapat di akses dengan *browser* yang berbeda-beda.
3. Jumlah *client* untuk pengujian di perbanyak.
4. Untuk meningkatkan kualitas, gunakan *hardware client* yang memadai. Perlu diperhatikan juga *software* yang berjalan secara *background* di sisi *client* seperti *antivirus* dan *software* yang melakukan update otomatis jika terkoneksi *internet*.
5. Untuk fasilitas *Public* pertimbangkan penggunaan *Pc Box / mikrotik router board* yang mempunyai RAM yang lebih besar dibanding *Access Point*

DAFTAR PUSTAKA

- Agustia, Dwi, Richi. “Rancang Bangun Media Informasi Kesenian Berbasis Web Dalam Bentuk Layanan Video On Demand (VOD) Dengan Menggunakan Metode Pseudo HTTP Streaming (Studi Kasus Bandung Heritage)”, *Video Streaming*. Hlm. 219-237, 2011.
- Akmal, Andrian. “Konfigurasi dan Analisis Manajemen Bandwidth pada PC Router Menggunakan Metode (Hierarchical Token Bucket) HTB dan Class Based Queue (CBQ)”, HTB vs CBQ. Hlm. 1-8, 2011.
- Bandung, Yohanes. “Analisis Kualitas Layanan Audio-Video Layanan Kelas Virtual di Jaringan Digital Learning Pedesaan”, *Quality of Service*. Hlm. 1-11, 2011.
- Chandra, H, Malora “*Action Research*” [Online] Available <http://chandrax.wordpress.com/2008/07/05/action-research-penelitian-tindakan>, diakses 20 Maret 2013.
- Davison, R. M., Martinsons, M. G., dan Kock Ned. “Journal : Information Systems Journal : Principles of Canonical Action Research”, *Action Research*. Hlm 14, 2004.
- Devara, Martin “*Hierarchical Token Bucket Theory*” [Online] Available <http://luxik.cdi.cz/~devik/qos/manual/theory.htm>, diakses 20 Februari 2013.
- Dwi, W, Ilham “*IEEE 802.11 Standar Wireless LAN*” [Online] Available <http://computerbroken.blogspot.com/2013/01/ieee-80211-standar-wireless-lan.htm>, diakses 20 Maret 2013.
- Editor “*Broadcast Quality Video Over Wireless LAN*” 2011. [Online] Available <http://solper.com/pic/Vol-16-hal-5-292.pdf>, diakses 20 Maret 2013.
- EL. “*Pengguna Internet Indonesia*” [Online] Available <http://www.gatra.com/iltek/internet/22070>, diakses 20 Februari 2013.

- Fatoni. "Analisis Kualitas Layanan Jaringan Intranet", *Quality of Service*. Hlm. 1-15, 2012.
- Irawati, Dyah, Indrarini. "Analisis Performansi IPTV pada Jaringan IPv4", *Quality of Service*. Hlm. E13-E20, 2011.
- Jonathan, Pradana, Antony. "Network Traffic Management, Quality of Service (Qos), Congestion Control dan Frame Relay", *QOS*. Hlm. 12-24, 2011.
- Kristianto, Dwi, Endi. "Menghitung Delay packet pada Jaringan menggunakan Wireshark", *Wireshark*. Hlm. 1-9, 2012.
- Pangera, A, Abbas. "Analisis Perbandingan HTB dan CBQ untuk Mengatur Bandwidth menggunakan Linux", *HTB vs CBQ*. Hlm. 1-14, 2000.
- Pradyani, A.P, Ida, dan Indrajaya, D.K, Kadek. "Arsitektur TCP/IP", *TCP/IP*. Hlm. 1-4, 2009.
- Pratania, Eka. "Analisa Perbandingan HTB dan CBQ untuk Manajemen Bandwidth pada Jaringan TCP/IP", *HTB vs CBQ*. 2003.
- Prihtiyani, Eny, dan Supriyadi, Marcus. "*Perkembangan Internet*" [Online]
Available <http://tekno.kompas.com/read/2012/04/17/14285429/Situs.Musik.Paling.Banyak.Diakses.htm>, diakses 20 Febuari 2013.
- Prihtiyani, Eny, dan Supriyadi, Marcus. "*Perkembangan Internet*" [Online]
Available <http://teknologi.kompasiana.com/internet/2013/01/07/18-19-juta-pengguna-baru-internet-di-2013-didominasi-kalangan-middle-class-517133.html>, diakses 20 Febuari 2013.
- Santosa, Budi. "Manajemen Bandwidth Internet dan Intranet", *HTB*. Hlm. 1-27, 2007.
- Sudiarjo, Slamet. "Pengukuran Parameter Kualitas Layanan (Qos) Trafik Video Streaming pada Jaringan IP berbasis Switch layer 2", *Video Streaming*. Hlm. 4-27, 2010.

TIPHON. "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON);General aspects of Quality of Service (QoS)", *QOS*. Hlm. 15-28, 1999.

Zahroh, Aminatul "*METODE PAR (PARTISIPATORY ACTION RESEARCH) DALAM PENELITIAN*" 2013. [Online] Available <http://gudangilmu-pendidikan.blogspot.com/2013/01/metode-penelitian-tindakan-kelas-ptk.html>, diakses 18 Juni 2013.